

Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ШАДУРА ОКСАНА ВІТАЛІЇВНА

УДК 004.942; 519.876.5

ДИСЕРТАЦІЯ
СТОХАСТИЧНА ОПТИМІЗАЦІЯ ПРОДУКТИВНОСТІ ПАКЕТУ
СИМУЛЯЦІЇ ТРАНСПОРТУ ЧАСТИНОК В ФІЗИЦІ ВИСОКИХ ЕНЕРГІЙ

05.13.12 – системи автоматизації проектувальних робіт

Подається на здобуття наукового ступеня кандидата технічних наук
Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

О.В. Шадура

Науковий керівник: Петренко Анатолій Іванович, доктор технічних наук,
професор

Київ – 2019

АНОТАЦІЯ

Шадура О.В. Стохастична оптимізація продуктивності пакету симуляції транспорту частинок в фізиці високих енергій. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.12 – «Системи автоматизації проектувальних робіт». – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Міністерство освіти і науки України, Київ, 2019.

Дисертаційна робота присвячена дослідженню методів оптимізації продуктивності пакету моделювання транспорту частинок у фізиці високих енергій GeantV за допомогою генетичних алгоритмів. Пакет GeantV розроблюється для вирішення задач обробки великих масивів даних в експериментах ATLAS, CMS, ALICE на Великому адронному колайдері (ВАК) в ЦЕРНі (м.Женева, Швейцарія) і є програмним пакетом функціонального аналізу (CAE-системи в САПР) фізичних явищ при транспорті частинок через середовище детектора. Перехід від використання гомогенного обчислювального середовища (грід) до гетерогенної обчислювальної інфраструктури з використанням суперкомп'ютерів для обробки очікуваного зростання обсягу масивів даних на наступних етапах роботи ВАК потребує розробки спеціальних методів оптимізації продуктивності і планування завдань моделювання фізичних явищ при обробці цих даних на основі паралельних обчислень.

З урахуванням цих викликів для аналізу масивів експериментальних даних на ВАК, розробка математичних моделей, методів оптимізації програмних додатків для CAE-системи автоматизованого проектування в фізиці високих енергій, а саме для оптимізації продуктивності пакету

моделювання (симуляцій) транспорту частинок GeantV, на основі аналізу, розробки та програмної реалізації модифікацій алгоритмів стохастичної оптимізації та використання методів машинного навчання для оптимізації налаштування симуляцій Монте-Карло є актуальними для розвитку методів функціонального аналізу в фізиці високих енергій (ФВЕ).

В дисертації використовується моделювання за допомогою принципу «чорного ящика», який зазвичай використовується у випадках, коли в процесі обробки даних постає задача підгонки даних, залежність яких складно визначити або які не можуть бути описані конкретною математичною моделлю. Для цього підходу існує набір інструментів, які пропонують декілька лінійних та нелінійних моделей для проблеми «чорного ящика», що традиційно використовується для динамічних систем. Структурні елементи цих моделей різняться залежно від гнучкості інструментів, необхідних для оцінки динаміки та шуму в системі.

Ефективність стратегій для оптимізації задачі «чорного ящика» зазвичай вимірюється кількістю проведених оцінок для досягнення оптимального значення або навпаки з погляду еволюції величини оцінки наближення до оптимального значення цільової функції для певної кількості оцінок.

Еволюційні алгоритми пропонують оптимізувати цільову функцію за допомогою процесу штучної еволюції для вибору найкращих параметрів. Наприклад, в генетичних алгоритмах значення параметрів розглядають як особини в поколіннях, які розвиваються від покоління до покоління в середовищі, що уможливлює виживання та відтворювання найпридатніших особин (найбільш придатними є найкращі параметри для цільової функції).

У дисертаційній роботі для наближеного розв'язку задач багатокритеріальної оптимізації розроблено оригінальний алгоритм на основі включення нового генетичного оператора (НГК-оператор), що побудований на основі методу нецентрованих головних компонент, який надає змогу

прискорити збіжність генетичного алгоритму до Парето-фронт.

Проведено порівняння використання методу головних компонент при його застосуванні для центрованих і нецентрованих даних, та показано еквівалентність використання обох представлень даних на основі оцінки середньоквадратичної похибки.

На основі запропонованого НГК-оператора розроблено програмний додаток та здійснена його інтеграція в бібліотеку генетичних алгоритмів програмного пакета GeantV, яка є спеціальною бібліотекою програмних додатків САЕ-системи автоматизованого проектування САД в фізиці високих енергій. Ця бібліотека буде використовуватись для оптимізації та збільшення ефективності використання обчислювальних ресурсів для моделювання та аналізу транспорту частинок через детектори у ФВЕ.

Знайдено оптимальну інтеграцію НГК-оператора в типовий генетичний алгоритм та досліджено ефективність його застосування на прикладі еталонних тестів для багатоцільових задач та у випадку оптимізації продуктивності пакету симуляцій транспорту частинок GeantV у порівнянні з неоптимізованою версією такого самого програмного пакета. Встановлено, що при запуску оптимізованого пакета у неоднорідному обчислювальному середовищі середній приріст продуктивності пакету досягає до 20% над неоптимізованим випадком.

В дисертаційній роботі розроблена бібліотека програмних модулів розширення функціональних можливостей генетичних алгоритмів для застосування в програмному пакеті GeantV, яка буде використовуватись для оптимізації продуктивності програмного пакета симуляції транспорту частинок GeantV при його застосуванні для обробки даних в експериментах на Великому адронному колайдері.

Новизна роботи полягає в наступному: вперше запропоновано математичну модель для методу нецентрованих головних компонент та отримано формулу для оцінки наближень при застосуванні методу для аналізу

даних, яка характеризується тим, що пришвидшує перевірку чутливості параметрів та забезпечує контролювання збереження максимуму інформації при цих наближеннях; розроблено математичну модель методу нецентрованих головних компонент, отримано формулу для оцінки наближень при застосуванні методу для аналізу даних; розроблено модифікація генетичного алгоритму завдяки введенню до стандартного набору операторів генетичних алгоритмів (селекція, мутація, кросовер) нового оператора, що визначається через нецентрований метод головних компонент (*R*-операція), і який діє на популяціях ГА. Використання методу головних компонент дає змогу перевірити чутливість параметрів генетичного алгоритму та можливу кореляцію між параметрами. Оцінка похибки наближення показує, що процедура редукції матриці даних на основі виділення власних векторів матриці нецентрованих других моментів для яких вона має найменші власні значення, є коректною. Це дає змогу припустити, що використання методу головних компонент для нецентрованої матриці даних для генетичних алгоритмів (ГА) може істотно прискорити процес пошуку Парето-фронт для задачі багатокритеріальної оптимізації.

Вперше запропоновано метод прискорення збіжності генетичного алгоритму до Парето-фронт для задач багатокритеріальної оптимізації. Метод характеризується тим, що частина популяції, яка переходить у наступне покоління, перевіряється на чутливість до негативного впливу «шуму» в генетичному алгоритмі і на можливу кореляцію між ними, та включається в генетичний алгоритм за допомогою нового генетичного оператора (НГК-оператору), що побудований на основі методу нецентрованих головних компонент.

Сформульовані рекомендації для застосування запропонованої в дисертаційній роботі модифікації генетичних алгоритмів для оптимізації пакетів симуляцій транспорту частинок, на прикладі пакету GeantV. Використання НГК-оператора дало змогу збільшити продуктивність

використання обчислювальних ресурсів для аналізу даних у фізиці високих енергій.

Результати, що отримані в ході цього дослідження, є доказом справедливості концепції оптимізації продуктивності GeantV з використанням еволюційних налаштувань для обчислень. Цей же метод може бути застосований для встановлення оптимізованих GeantV-додатків на суперкомп'ютерах і кластерах для високопродуктивних обчислень, одночасно з можливістю налаштування паралельних обчислень для наборів задач, оброблених у неоднорідному середовищі, а також для досягнення масштабованості у середовищах для високопродуктивних обчислень.

Ключові слова: генетичний алгоритм, динамічна система, нерухомі точки, стохастична оптимізація, генетичний оператор, аналіз основних компонентів, оптимізація продуктивності.

ABSTRACT

Shadura O.V. Stochastic optimization of the performance of a particle transport simulation package in High Energy Physics. – Dissertation for a Degree of Candidate of Technical Sciences (PhD), specialization 05.13.12 – «Automation of systems for computer aided design». – National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Ministry of science and education of Ukraine, Kyiv, 2019.

The dissertation is dedicated to study the optimization strategies using genetic algorithms to achieve the best computing performance of the particle transport simulation package GeantV for the high energy physics (HEP) simulations. It had been developed to solve the problems of processing the large data sets in the ATLAS, CMS, ALICE experiments at the Large Hadron Collider (LHC), located in CERN, Switzerland, Geneve. The transition from a homogeneous computing environment (grid computing) to a heterogeneous computing infrastructure, using supercomputers to handle the expected growth of data volumes in the next stages of LHC, requires the development of specific methods for the performance optimization and complex scheduling tasks while performing parallel data analysis in HEP.

Given these challenges, we introduced novel mathematical algorithms and software application, used to optimize the performance of the GeantV particle transport simulation package. One of the proposed approaches is to develop an algorithm based on the stochastic optimization (genetic algorithms) and machine learning methods to optimize the tuning of Monte Carlo simulations and which could be applicable for the development of new applications and other technologies in HEP.

Black box modeling is commonly used in the data processing when it comes to the complex task of fitting data and where dependencies are difficult to

determine or they cannot be described by a specific mathematical model. This approach can also be used in the cases, where instead of the defined properties and interconnections of system components in the mathematical model, it is necessary to study the response of the whole system. Selecting this approach, we developed a model of genetic algorithm, representing it as a dynamic system. The structural elements of this model vary depending on the flexibility of the model and could evaluate different dynamics of multiple complex processes and the noise of the system.

The efficiency of evolutionary strategies used for optimisation of the black box problems is usually measured by the number of evaluations performed to achieve the optimal value, or vice versa, in terms of the evolution of the magnitude of the approximation to the optimal value of the objective function for a given number of estimates.

Evolutionary algorithms propose to optimize the fitness function through an artificial evolution process to be able to select the best parameters. For example, in this case, values of parameters are considered as individuals in genetic generations, evolving from one genetic generation to other generation in an environment that makes it possible to survive and reproduce the most suitable individuals (in this case the best fit for the fitness function is considered to be the most appropriate).

In this dissertation, as a part of solution of multi-criteria optimization problem, was proposed an additional genetic operator (Uncentered Principle Component Analysis operator (UPCA operator)). It is using the non-centered principal component analysis method, which allows accelerating the convergence of the genetic algorithm population closer to the optimal Pareto front.

Comparison of the principal component analysis method for centered and uncentered data and equivalence of its usage for both representations had been presented in the scope of this dissertation and was based on the estimation of the mean square error.

Based on the idea of proposed UPCA operator, it has developed a software application (e.g. a library of genetic algorithms developed for the GeantV software package) and it was integrated into a computer-aided design system GeantV. It will be used for the optimisation and increasing the efficiency of the usage of computational resources for the data analysis in High Energy Physics.

The optimal integration of the UPCA operator in a typical genetic algorithm also has been discussed in the scope of this thesis. The efficiency of the algorithm has been tested using generic multi-objective optimization benchmark tests. Simple performance benchmarks using the GeantV particle transport simulation package was compared versus its non-optimized version and its comparisons are also provided in the scope of this thesis. The results shown that the average productivity was increased up to 20% over the non-optimized package, when tested in the non-homogeneous computing environment.

In the scope of this dissertation had been developed a software library integrating the genetic algorithms as a part of the GeantV software package. It will be used to optimize the performance of GeantV software package for the particle transport simulation, when used for processing the physics data, collected by experiments on the Large Hadron Collider (LHC).

The mathematical model of the non-centered principal components analysis method is defined together with the formula for estimation of approximations for the data analysis. The estimation of the approximation error shows that the procedure of reducing the data matrix, based on the selection of eigenvectors of the matrix of non-centered second moments for which it has the smallest eigenvalues, is correct. This suggests that the use of the principal component analysis method for a non-centered data matrix in genetic algorithms can significantly accelerate the Pareto front search process for multicriteria optimization problem.

A modification of the genetic algorithm is defined by introducing into the standard set of genetic algorithm operators (selection, mutation, crossover), a new operator, which is determined by the non-centered principal component analysis

method (new genetic operator to be used on genetic populations). The use of the principal component method makes it possible to check the sensitivity of the genetic algorithm parameters and the possible correlation between the parameters.

For the first time, it was developed a mathematical model for the uncentered principal component analysis method. Proposed method is characterized by the significant acceleration of the simulation's parameter sensitivity detection. It provides an optimal control of the retention of maximum amount of information after data conversion/approximation.

The method of accelerating the convergence of the genetic algorithm to the optimal Pareto front for multi-criteria optimization problems has been proposed in this work. The proposed method is using the idea that a part of the genetic population, which will be used by the next genetic generation (population) of genetic algorithm, is tested for the sensitivity to the negative influence of existence «noise» in the genetic algorithm population and a possible correlation between data. This step was adopted in the genetic algorithm by an introduction of a new genetic operator (UPCA operator), which had developed using the non-centered principal component analysis method.

Proposed modification of genetic algorithms for the performance optimization of particle transport simulations, such as the GeantV simulations, have been defined in the scope of this thesis. The usage of a proposed UPCA operator together with genetic algorithms increased the productivity of utilization of computing resources used in High Energy Physics data analysis.

The results of this study are presenting the proof of the concept of optimization of GeantV performance using evolutionary algorithms. The same method can be used to deploy GeantV applications on supercomputers and clusters for the efficient high-performance computing, while configuring massively parallel GeantV simulations, launched in a non-homogeneous computing environment and providing optimal scalability in high-performance computing environment.

Keywords: genetic algorithm, dynamic system, fixed points, stochastic optimization, genetic operator, principle component analysis, performance optimization.

Список опублікованих праць за темою дисертації

Основні публікації

1. Shadura O. Multivariate convergence-targeted operator for the genetic algorithm / O. Shadura, A. Petrenko, S. Svistunov // Системні дослідження та інформаційні технології. – 2017. – № 1. – С. 126 – 140. – Бібліогр.: 17 назв.
2. Оксана Шадура. Метод головних компонент і оптимізація пакетів фізичного моделювання за допомогою генетичних алгоритмів // Вісник Університету «Україна», Серія «Інформатика, обчислювальна техніка та кібернетика». – 2019. – №1 (22) – С. 198 – 209. – Бібліогр.: 16 назв.
3. Оксана Шадура. Модифікація генетичних алгоритмів на основі методу нецетрованих головних компонент та стандартні тести, // World Science – 2019. – № 4(44). – С. 4 – 11. – Бібліогр.: 10 назв. – DOI: 10.31435/rsglobal_ws.
4. Oksana Shadura. Performance Optimization of Physics Simulations Through Genetic Algorithms. / Oksana Shadura, Federico Carminati and Anatoliy Petrenko. // Journal of Computer Science. – 2019. – Volume 15. – Issue 1. – P. 57–66. – DOI 10.3844/jcssp.2019.57.66.
5. Оксана Шадура. Оптимізація пакету фізичного моделювання GeantV за допомогою генетичних алгоритмів. // Вчені записки Таврійського національного університету імені В. І. Вернадського, Серія: Технічні науки – 2019. – Том 30 (69). – № 3. – С. 206–213.

Апробація результатів дисертації

6. Oksana Shadura et al. The experimental facility for the Search for Hidden Particles at the CERN SPS / Ahdida C., Albanese R., ..., Shadura O., ... , Zaytsev

Yu // Journal of Instrumentation (JINST). – 2019. – Volume 14. – Issue 3. – P. 3025–3046 – DOI 10.1088/1748–0221/14/03/p03025

7. Oksana Shadura. Stochastic performance tuning of complex simulation applications using unsupervised machine learning. // Oksana Shadura, Federico Carminati // 2016 IEEE Symposium Series on Computational Intelligence – 2016. – ISBN 978-1-5090-4240-1.

8. G. Amadio and all. Stochastic optimization of GeantV code by use of genetic algorithms / G. Amadio and J. Apostolakis and M. Bandieramonte and S.P. Behera and R. Brun and P. Canal and F. Carminati and G. Cosmo and L. Duhem and D. Elvira and G. Folger and A. Gheata and M. Gheata and I. Goulas and F. Hariri and S.Y. Jun and D. Konstantinov and H. Kumawat and V. Ivantchenko and G. Lima and T. Nikitina and M. Novak and W. Pokorski and A. Ribon and R. Sehgal and O. Shadura and S.Vallecora and S. Wenzel, Stochastic optimization of GeantV code by use of genetic algorithms // Journal of Physics: Conf. Ser. Offline Computing – 2017. – Vol. 898, N. 4, – P. 042026 – IOP Publishing.

9. Amadio G and all. GeantV: from CPU to accelerators / G. Amadio, A. Ananya, J. Apostolakis, A. Arora, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F. Carminati, L. Duhem, D. Elvira, A. Gheata, M. Gheata, I. Goulas, R. Iope, S. Jun, G. Lima, A. Mohanty, T. Nikitina, M. Novak, W. Pokorski, A. Ribon, R. Sehgal, O. Shadura, S. Vallecora, S. Wenzel, Y. Zhang // Journal of Physics: Conf. Ser. Computing Technology for Physics Research – 2016. – Vol. 762, N. 1. – P. 012019. – IOP Publishing.

10. Amadio G and all. The GeantV project: preparing the future of simulation // G. Amadio, J. Apostolakis, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F. Carminati, J. C. De Fine Licht, L. Duhem, V. D. Elvira, A. Gheata, R. L. Iope, G. Lima, A. Mohanty, T. Nikitina, M. Novak, W. Pokorsky, R. Sehgal, O. Shadura, S. Vallecora, S. Wenzel // Journal of Physics: Conf. Ser. Offline software – 2015. – Vol. 664, N. 7. – P. 072006 – IOP Publishing.

11. Apostolakis J and all. Adaptive track scheduling to optimize concurrency

and vectorization in GeantV // J.Apostolakis, M.Bandieramonte, G. Bitzes, R. Brun, P. Canal, F Carminati, J. C. De Fine Licht, L. Duhem, V. D. Elvira, A. Gheata, S. Y. Jun, G. Lima, T. Nikitina, M. Novak, R. Sehgal, O. Shadura, S. Wenzel // Journal of Physics: Conf. Ser. Advanced Computing and Analysis Techniques in Physics Research – 2015. – Vol. 608, N 1. – P. 012003 – IOP Publishing.

12. J. Apostolakis and all. Towards a high performance geometry library for particle–detector simulations // J. Apostolakis and M. Bandieramonte, G. Bitzes and R. Brun, P. Canal, F.Carminati, G. Cosmo, J. C. De Fine Licht, L. Duhem, V. D. Elvira, A. Gheata, S. Y. Jun, G. Lima, T. Nikitina, M. Novak, R. Sehgal, O. Shadura, S. Wenzel // Journal of Physics: Conf. Ser. Computing Technology for Physics Research – 2015. – Vol. 608, N. 1, – P. 012023 – IOP Publishing.

ЗМІСТ

ВСТУП	20
РОЗДІЛ І. СТРУКТУРНІ ЕЛЕМЕНТИ ТА АЛГОРИТМИ БАГАТОКРИТЕРІАЛЬНОЇ ЗАДАЧІ ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ ПАКЕТУ МОДЕЛЮВАННЯ ТРАНСПОРТУ ЧАСТИНОК GEANTV	30
1.1 Великий адронний колайдер – найбільший та найпотужніший у світі прискорювач елементарних частинок	30
1.1.1 БАК з високою яскравістю	31
1.2 Експерименти БАК: CMS, ATLAS, ALICE, LHCb	32
1.2.1 Експеримент БАК CMS	32
1.2.2 Експеримент БАК ATLAS	34
1.2.3 Експеримент БАК ALICE	35
1.2.4 Експеримент БАК LHCb	35
1.3 Інші не-БАК експерименти	36
1.4 Обчислювальні ресурси, що використовуються для аналізу даних БАК	36
1.4.1 Грід-обчислення	37
1.4.2 Хмарні обчислення	38
1.4.3 Волонтерські ресурси	39
1.5 Поточний стан оптимізації ефективності обчислень у ФВЕ	39
1.5.1 Оцінка ефективності ресурсів	42
1.6 Опис САЕ-системи моделювання транспорту частинок GeantV ..	44
1.7 Теорія та алгоритми, що використовуються в дослідженні	47
1.7.1 Задача багатокритеріальної оптимізації	50

1.7.2	Визначення оптимуму Парето	51
1.8	Генетичні алгоритми як спеціальний випадок алгоритмів стохастичної оптимізації	53
1.8.1	Генетичний алгоритм з недомінантним сортуванням – NSGA-II/III.....	57
1.8.2	Генетичний алгоритм з недомінантним сортуванням – NSGA-II..	57
1.8.3	Генетичний алгоритм з недомінантним сортуванням – NSGA-III	58
1.9	Оптимізація «чорного ящика» як динамічна система. Теорія генетичних алгоритмів.....	60
1.9.1	Марківські ланцюжки та генетичні алгоритми.....	61
1.9.2	Проста модель генетичного алгоритму	62
1.10	Особливі точки та їх роль в генетичному алгоритмі.....	68
1.10.1	Поведінка вихідної популяції для скінченних популяцій	69
1.10.2	Фіксовані точки та локальні оптимуми	69
	Висновки до розділу 1	71
	РОЗДІЛ 2. МЕТОД НЕЗЦЕНТРОВАНИХ ГОЛОВНИХ КОМПОНЕНТ ЯК ОСНОВА ДЛЯ НОВОГО ОПЕРАТОРА В ГЕНЕТИЧНИХ АЛГОРИТМАХ	73
2.1	Редукція цільових функцій в задачах багатокритеріальної оптимізації.....	74
2.1.1	Метод «збереження відношення домінування».....	75
2.1.2	Еволюційний алгоритм пошуку кута Парето.....	76
2.1.3	Спосіб зменшення кількості цільових функцій на основі методу головних компонент	77
2.2	Метод головних компонент	78
2.3	Метод головних компонент для центрованої матриці даних.....	81

2.3.2 Підхід, що базується на сингулярному представленні матриці даних.....	86
2.4 Метод головних компонент для незцентрованої матриці даних	90
2.4.1 Підхід, що базується на сингулярному представленні матриці даних.....	94
2.5 Оператор редукції цільових параметрів в генетичних алгоритмах для багатоблочних паралельних застосувань	97
Висновки до розділу 2	100
РОЗДІЛ 3. РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ НГК-ОПЕРАТОРА ДЛЯ ЕТАЛОННИХ ТЕСТІВ БАГАТОЦІЛЬОВИХ ЗАДАЧ	102
3.1 Стандартні методи перевірки нових алгоритмів та набір тестів DTLZ	102
3.2 Вибір генетичного алгоритму	111
3.3 Оптимізація схеми генетичного алгоритму	113
3.3.1 Метод NSGA-II - UPCA1 та включення НГК-оператора в генетичний алгоритм	114
3.3.2 Метод NSGA-II - UPCA2 включення НГК-оператора в генетичний алгоритм	117
3.3.3 Метод NSGA-II - UPCA3 включення НГК-оператора в генетичний алгоритм	118
3.3.4 Метод NSGA-II - UPCA4 включення НГК-оператора в генетичний алгоритм	119
3.3.5 Приклади розв'язання типових задач багатокритеріальної оптимізації.....	120
Висновки до розділу 3	126
РОЗДІЛ 4. ТЕСТУВАННЯ ЗАСТОСУВАННЯ НГК-ОПЕРАТОРА В ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ ПАКЕТУ МОДЕЛЮВАННЯ	

ТРАНСПОРТУ ЧАСТИНОК GEANTV	127
4.1 Програмний пакет Geant4	127
4.1.1 Geant4 та експеримент SHiP	129
4.1.2 Оптимізація мюонного щита для SHiP	130
4.1.3 Оцінювання рівня сигналу та фону в експерименті SHiP	133
4.2 GeantV – наступне покоління програмних пакетів для моделювання транспорту частинок.....	137
4.2.1 Визначення багатокритеріальної задачі для моделювання з використанням CAE-системи GeantV	139
4.2.2 Огляд прототипу GeantV	141
4.2.3 Модель паралелізму GeantV	147
4.2.4 Ключові елементи основної інфраструктури GeantV	149
4.2.5 Налаштування параметрів для багатокритеріальної задачі оптимізації CAE-системи GeantV.....	151
4.3 Високопродуктивні розрахунки у моделюванні за допомогою CAE-системи GeantV та їх обчислювана складність	152
4.4 Масивні паралельні обчислення і очікувані результати продуктивності.....	154
4.5 Тестові налаштування GeantV.....	159
4.6 Результати тестування	160
Висновки до розділу 4	164
ВИСНОВКИ	166
ПЕРЕЛІК ПОСИЛАНЬ	169
Додаток А. СХЕМА АЛГОРИТМУ NSGA-II	182
Додаток Б. ПОРІВНЯННЯ АЛГОРИТМІВ NSGA-II ТА NSGA-II-UPCA.....	183

Додаток В. АКТИ ВПРОВАДЖЕННЯ.....	201
Додаток Г. СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ	204

ВСТУП

Актуальність теми. На теперішній час існує протиріччя між вимогами до засобів автоматизації проектування складних об'єктів в фізиці високих енергій та сучасним станом методології та засобів оптимізації продуктивності обчислень у САЕ-системах автоматизованого проектування (CAD). Для його розв'язання була сформульована наукова-прикладна задача що дозволяє дослідити оптимізацію продуктивності програмних додатків для САЕ-системи автоматизованого проектування (CAD) в фізиці високих енергій використовуючи алгоритми стохастичної оптимізації на прикладі оптимізації продуктивності пакету моделювання (симуляцій) транспорту частинок GeantV.

GeantV є прототипом програмного забезпечення наступного покоління для моделювання транспорту частинок через різні типи матеріалів і має увібрати в себе та розвинути всі позитивні сторони САЕ-системи автоматизованого проектування і розрахунків Geant4, яка безумовно є стандартом для обробки даних у фізиці високих енергій (ФВЕ), ядерній фізиці, фізиці прискорювачів, медицині та фізиці космосу, і яку, зокрема використовують для аналізу даних в експериментах на Великому адронному колайдері (ВАК), розташованому в ЦЕРН, м. Женева, Швейцарія.

Не дивлячись на багаточисленні публікації по цій темі, в результаті її аналізу було знайдено протиріччя між необхідністю збільшення ефективності існуючих підходів і обмеженість математичних моделей та методів розв'язання задач оптимізації систем проектування.

Основна проблематика полягає у тому що експерименти на прискорювачі ВАК генерують велику кількість даних, обробка яких вимагає більш ефективного використання обчислювальної техніки нового покоління, що потребує розробки нових підходів до методів розв'язання

задач оптимізації систем проектування. Основними особливостями цих задач є:

- наявність великого та не гетерогенного пулу ресурсів, що задіяний для аналізу даних на прискорювачі ВАК та вимагає додаткових зусиль для оптимізації продуктивності і планування завдань проведення паралельних обчислень при обробленні даних;
- алгоритмізація завдань проектування за допомогою методів машинного навчання, які надають можливість ефективніше використовувати ресурси та проводити подальше прогнозування оптимальних параметрів для конкретного обчислювального середовища.

З урахуванням того що, для аналізу даних в експериментах на Великому адронному колайдері (ВАК) використовують ґрід-технології, які розподіляють обробку експериментальних даних серед великої кількості окремих обчислювальних кластерів, що побудовані з різних за структурою комп'ютерних платформ, у цьому випадку пріоритет полягає у використанні контрольованого машинного навчання для класифікації подій для об'єкта реконструкції.

Важливим зауваженням є те, що для ефективного аналізу ми маємо забезпечити ефективну стадію попереднього навчання. У випадку оптимізації використання обчислювальних ресурсів для аналізування даних, нам потрібно буде приймати рішення в режимі реального часу та безпосередньо вивчати нові шаблони даних під час проведення аналізу. Задача оптимізації продуктивності програмних додатків для САЕ-системи автоматизованого проектування (CAD) в фізиці високих енергій є ідеальним випадком використання методів неконтрольованого навчання, поєднаного із стохастичними методами оптимізації.

Додавання спеціальних операторів до стохастичних методів оптимізації (генетичних алгоритмів) може привести до зменшення шуму при виборі вдалого наближення та більш швидкого зближення з фронтом Парето, який

визначає набір оптимальних розв'язків, що не домінують відносно один одного, та при переміщенні від однієї точки фронту Парето до іншої досягається певний програш в одній фітнес-функції та виграш в іншій. Фронтом Парето складається з ідеальних осіб популяції в генетичному алгоритмі (ГА), що вибрані на основі відповідного набору параметрів оптимізації, і для його знаходження зручно застосувати ортогональні перетворення та розглядати властивості варіації даних, щоб виявити специфічні структури в даних.

Зокрема під час дослідження компонентів системи проектування було проведено ідентифікацію та дослідження параметрів для знаходження їх оптимальних значень, наприклад таких як час роботи, об'єм споживання пам'яті, тощо, що є важливими для налаштування продуктивності системи автоматизованого проектування GeantV для обробки експериментальних даних.

Запропонований в роботі метод дає змогу розв'язати задачу оптимального налаштування параметрів симуляції транспорту частинок з урахуванням «часу виконання» паралельних обчислень на декількох типах комп'ютерних платформ, що зробить більш швидшим типовий аналіз даних методом Монте-Карло для експериментів у БАК.

Обґрунтування вибору теми дослідження. Дослідження в області ФВЕ та ядерної фізики неможливі без використання великих обчислювальних потужностей та спеціального програмного забезпечення для обробки, моделювання та аналізу даних. Така ситуація обумовлена як великою кількістю даних, що генерується в експериментах на сучасних прискорювачах, так і статистичною природою аналізу даних і складністю алгоритмів обробки даних, які використовуються.

Введення в дію Великого адронного колайдера (БАК) в ЦЕРНі, м. Женева та експлуатація детекторів в експериментах ATLAS, CMS, ALICE потребують розробки та застосування нових підходів, методів і рішень в

області інформаційних технологій. В нинішньому режимі роботи прискорювача кожні 50 наносекунд відбувається зіткнення протонів при енергії 13 ТэВ з народженням близько 1600 елементарних частинок, які реєструються та аналізуються тригером високого рівня. Статистика, що набрана за час роботи LHC в 2010-2017 рр., складає понад 60 Пбайт «сирих» (необроблених) даних. Безпрецедентний обсяг інформації, що надходить вже зараз, і очікуване зростання обсягу даних на наступних етапах роботи колайдера, потребують розробки нових комп'ютерних моделей, методик та методів управління роботи з даними, створення нових програмних систем для обробки даних.

Одним з підходів для вирішення задачі обробки великих масивів даних на ВАК став перехід від використання гомогенного обчислювального середовища (грід) до гетерогенної обчислювальної інфраструктури з використанням суперкомп'ютерів. Це потребує розробки спеціальних методів оптимізації продуктивності для існуючих CAE-систем автоматизованого проектування в фізиці високих енергій і планування завдань при проведенні паралельних обчислень при обробці даних, як наприклад для застосування нового програмного пакету GeantV, який є прототипом програмного забезпечення наступного покоління для моделювання транспорту частинок через різні типи матеріалів.

Він зберігає всі позитивні сторони CAE-системи автоматизованого проектування і розрахунку Geant4, що є стандартом для обробки даних в ФВЕ, ядерній фізиці, в фізиці прискорювачів, медицині та фізиці космосу. Іншим важливим підходом є розвиток нових алгоритмічних методів, пов'язаних з машинним навчанням, які надають можливість ефективніше використовувати ресурси та проводити подальше прогнозування оптимальних параметрів завантаження для конкретного обчислювального середовища.

З урахуванням цих викликів для аналізу масивів експериментальних даних на ВАК, розробка математичних моделей, методів проектування систем

та нових програмних додатків для існуючих САЕ-систем автоматизованого проектування в фізиці високих енергій, а саме для оптимізації продуктивності пакету моделювання транспорту частинок GeantV, на основі аналізу, розробки та програмної реалізації модифікацій генетичних алгоритмів оптимізації та використання методів машинного навчання для оптимізації налаштування симуляцій Монте-Карло є актуальними для розвитку методів та задач проектування в фізиці високих енергій.

У цьому напрямку вагомі результати отримали І.Фостер (I.Foster), К.Кессельман (C.Kesselman), Ф.Корбато (F.Corbato), А.І. Петренко, С.Я. Свістунов, П.В. Свірін та інші.

Зв'язок роботи з науковими програмами, планами, темами. Дисертацію виконано згідно плану досліджень науково-дослідної роботи кафедри системного проектування НТУУ «КПІ імені Ігоря Сікорського» за темою «Дослідження нової концепції побудови динамічної архітектури проблемно-орієнтованого програмного забезпечення в грід-хмарному середовищі з елементами пост бінарних обчислень» (№ 2710-ф, номер державної реєстрації 0114U003449, терміни виконання: 2014-2016 рр.) та згідно плану досліджень у рамках програми CERN Doctoral Student Programme 2014-2017, відділу SoFTware at Physics Department (EP-SFT), ЦЕРН, Женева, Швейцарія.

Мета та завдання дослідження. Метою дисертаційної роботи є розробка математичних моделей та методів оптимізації продуктивності САЕ-систем автоматизованого проектування для ФВЕ для збільшення продуктивності пакету моделювання транспорту частинок GeantV за допомогою генетичних алгоритмів оптимізації та їх програмної реалізації. Досягнення цієї мети полягає у вирішенні наступних задач:

- провести огляд і аналіз сучасного стану проблеми оптимізації продуктивності САЕ-систем автоматизованого проектування;
- розробити математичну модель для оптимізації роботи САЕ-системи

моделювання транспорту частинок GeantV для обробки даних на ВАК на основі генетичних алгоритмів оптимізації багатокритеріальних задач;

- виконати формалізацію методу нецентрованих головних компонент та знаходження оцінки наближення при його застосуванні до аналізу даних.

- розробити модель методу нецентрованих головних компонент та метод оцінки наближень при застосуванні методу для аналізу даних;

- розробити алгоритми та програмне забезпечення для вирішення завдань включення в генетичний алгоритм нового оператора генетичного алгоритму на основі методу нецентрованих головних компонент (НГК-оператор);

- перевірити ефективність застосування нового алгоритму для еталонних тестів багатокритеріальних задач;

- провести порівняльний аналіз ефективності включення НГК-оператора в генетичний алгоритм для оптимізації продуктивності САЕ-системи моделювання транспорту частинок GeantV в порівнянні з іншими реалізаціями генетичних алгоритмів.

Об'єктом теоретичних і експериментальних досліджень відповідно до поставленої мети є система автоматизованого проектування і розрахунку симуляції транспорту частинок GeantV.

Предметом дослідження дисертаційної роботи є процес стохастичної оптимізації продуктивності САЕ-системи симуляцій транспорту частинок GeantV на основі модифікації генетичних алгоритмів за допомогою методу нецентрованих головних компонент.

Методи дослідження. Для розв'язку задачі стохастичної оптимізації продуктивності САЕ-системи моделювання транспорту частинок GeantV в фізиці високих енергій використовувалися методи математичного моделювання, штучного інтелекту та інженерії програмного забезпечення, апарат теорії ймовірності та математичної статистики.

Методами досліджень є:

- генетичні алгоритми для оптимізації продуктивності системи автоматизованого проектування і розрахунку симуляції транспорту частинок GeantV;

- представлення генетичного алгоритму як динамічної системи на основі його зв'язку з марковськими ланцюжками;

- метод нецентрованих головних компонент;

- модифікація генетичного алгоритму за допомогою введення нового НГК-оператора в динамічну систему генетичного алгоритму.

Наукова новизна отриманих результатів.

Вперше:

- запропоновано математичну модель для оптимізації роботи САЕ-системи моделювання транспорту частинок GeantV для обробки даних у ВАК на основі генетичних алгоритмів для розв'язання типових багатокритеріальних задач;

- запропоновано математичну модель методу нецентрованих головних компонент та отримано оцінку наближень при застосуванні методу для аналізу даних, який характеризується тим, що пришвидчує перевірку чутливості параметрів та забезпечує збереження максимуму інформації при цих наближеннях;

- запропоновано метод прискорення збіжності генетичного алгоритму до Парето-фронту для розв'язання задач багатокритеріальної оптимізації;

- розроблено застосування запропонованого методу прискорення збіжності генетичного алгоритму до оптимального Парето-фронту на основі модифікації генетичних алгоритмів для оптимізації продуктивності програмного пакету САЕ-системи моделювання транспорту частинок GeantV в фізиці високих енергій.

Практичне значення отриманих результатів:

- розроблений в дисертаційній роботі оригінальний алгоритм для включення нового генетичного оператора (НГК-оператор), що побудований

на основі методу нецентрованих головних компонент, надає змогу прискорити збіжність генетичного алгоритму до оптимального фронту Парето і тим самим прискорити отримання наближеного рішення для задач багатокритеріальної оптимізації;

- проведено порівняння використання методу головних компонент при його застосуванні для центрованих і нецентрованих даних, та показано еквівалентність використання обох представлень даних на основі оцінки середньоквадратичної похибки;

- знайдено оптимальну інтеграцію НГК-оператора в типовий генетичний алгоритм та досліджено ефективність його застосування на прикладі еталонних тестів для багатокритеріальних задач;

- розроблена в дисертаційній роботі бібліотека програмних модулів для застосування в САЕ-системі моделювання транспорту частинок GeantV буде використовуватись для оптимізації продуктивності програмного пакету GeantV при його застосуванні для обробки даних в експериментах на Великому адронному колайдері.

Практичне значення результатів підтверджується їх впровадженням.

Розроблені в дисертаційній роботі математичні моделі, методи та програмне забезпечення для вирішення завдань структурно впроваджені: в держбюджетні науково-дослідні роботи, виконувані в рамках тематичних планових науково-дослідних робіт у Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського», що фінансуються за рахунок Міністерства освіти і науки; в Інституті теоретичної фізики ім. М.М. Боголюбова НАН України (додаток В).

Особистий внесок здобувача. Наукові положення, висновки і рекомендації, що виносяться на захист, отримано здобувачем самостійно. Із наукових праць, опублікованих у співавторстві, у дисертації використані лише ті ідеї, положення та матеріали, які є результатом власних досліджень автора. У роботах, написаних у співавторстві, автору належать наступні

результати: у [89] розроблено математичний метод нецентрованих головних компонент (НГК), отримано оцінку наближень при застосуванні методу для аналізу даних та запропоновано модифікацію генетичних алгоритмів за допомогою НГК-оператора для оптимізації пакетів фізичного моделювання; у [90] запропоновано модифікацію генетичних алгоритмів за допомогою НГК-оператора для оптимізації пакетів фізичного моделювання; проведений аналіз вибору оптимальної модифікації генетичного алгоритму за рахунок включення в алгоритм НГК-оператора та досліджена продуктивність такого генетичного алгоритму на наборі стандартних тестів DTLZ для оптимізації багатокритеріальних задач; у [91] сформульовано підхід до оптимізації системи автоматизованого проектування і розрахунку GeantV, що спрямований на розробку спеціальних методів оптимізації продуктивності та планування завдань при проведенні паралельних обчислень при обробці даних з застосуванням цієї системи автоматизованого проектування і розрахунку; у [13, 15, 20, 92, 104] досліджено вибір ключових параметрів системи автоматизованого проектування і розрахунку GeantV для оптимізації продуктивності цієї системи автоматизованого проектування і розрахунку та перевірено ефективності цього набору параметрів при дослідженні застосування оптимізованої системи автоматизованого проектування і розрахунку GeantV в порівнянні з неоптимізованим пакетом на прикладі обробки експериментальних даних на різних процесорах Intel (R) Core i7 та Intel (R) Ivy Bridge; у [16, 105] показано, що модифікований генетичний алгоритм з включенням НГК-оператора прискорює оптимізацію продуктивності системи автоматизованого проектування і розрахунку GeantV при симуляції транспорту частинок через простий калориметр та збільшує ефективність використання обчислювальних ресурсів для аналізу даних у фізиці високих енергій; у [6] проведено аналіз слабких та сильних сторін системи автоматизованого проектування і розрахунку Geant4, яка є попередником системи автоматизованого проектування і розрахунку нового

покоління GeantV, на основі аналізу даних в експерименті SHiP.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідалися, обговорювалися та отримали схвальну оцінку на міжнародних конференціях та семінарах: IEEE Symposium Series on Computational Intelligence, SSCI 2016, (Афіни, Греція, 2016, 2016); Міжнародна конференція «Computing in High Energy Physics, CHEP 2016» (Сан-Франциско, США, 2016); Міжнародна конференція «Computing in High Energy Physics, CHEP 2015», (Окінава, Японія, 2016); «16-й Міжнародний семінар з передових обчислювальних та аналітичних методів у фізиці», ACAT 2014, (Прага, Чеська Республіка, 2014).

Обсяг і структура дисертації. Дисертаційна робота викладена на 208 сторінках машинописного тексту, складається зі вступу, 4 розділів, загальних висновків, списку використаних джерел та 4 додатка. Обсяг основного тексту дисертації складає 168 сторінок друкованого тексту. Робота ілюстрована 66 рисунками. Список використаних джерел містить 121 найменування, з них 3 найменування кирилицею та 118 найменувань латиницею.

РОЗДІЛ I

СТРУКТУРНІ ЕЛЕМЕНТИ ТА АЛГОРИТМИ

БАГАТОКРИТЕРІАЛЬНОЇ ЗАДАЧІ ОПТИМІЗАЦІЇ

ПРОДУКТИВНОСТІ ПАКЕТУ МОДЕЛЮВАННЯ ТРАНСПОРТУ

ЧАСТИНОК GEANTV

1.1 Великий адронний колайдер – найбільший та найпотужніший у світі прискорювач елементарних частинок

Великий адронний колайдер (ВАК) (рисунок 1.1) [55] – це прискорювач з двома кільцями, побудований Європейською організацією з ядерних досліджень (ЦЕРН) і розташований на франко-швейцарському кордоні поблизу міста Женева у Швейцарії, на місці де був розташований попередній великий електрон-позитронний колайдер (LEP).

Мета ВАК полягає в тому, щоб дати вченим експериментальну установку, яка дозволила б їм перевірити сучасні теоретичні концепції в фізиці високих енергій, наприклад, існування бозона Хіггса та суперсиметрії. Прикладом одного з перших важливих результатів на ВАК було відкриття нової частинки, бозону Хіггса, що було оприлюднено 4 липня 2012 року і який був передбачений Стандартною моделлю елементарних частинок.

На ВАК працюють чотири експериментальні колаборації: CMS, ATLAS, ALICE, LHCb, кожна з яких має свій детектор для виконання експериментальних досліджень у фізиці високих енергій.

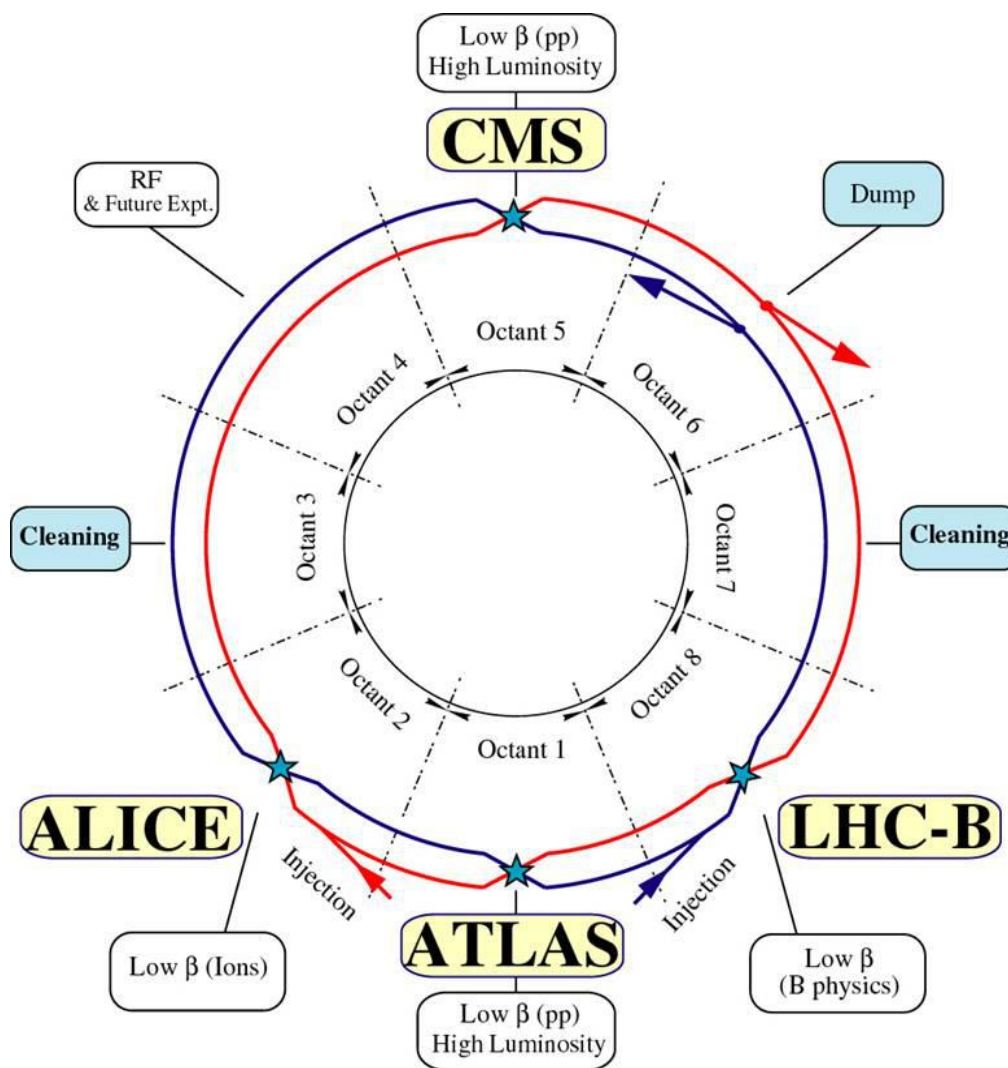


Рис. 1.1 – Великий адронний колайдер

1.1.1 ВАК з високою яскравістю

ВАК з високою яскравістю (англ. High Luminosity Large Hadron Collider (HL-LHC)) [19] є модернізацією ВАК для досягнення миттєвої яскравості яка в п'ять разів більша, ніж номінальна величина ВАК. Це дозволяє експериментальним колабораціям збільшити свій вибір даних, що на порядок більше в порівнянні з базовою програмою ВАК. Після п'ятирічного дослідження та розробки ідеї, цей складний проект вимагає близько десяти років розробок, прототипування, тестування та впровадження, і очікується,

що його функціонування почнеться в середині наступного десятиліття. Хронологія проекту продиктована тим фактом, що на початку наступного десятиліття багато критичних компонентів прискорювача досягне кінця свого життя через радіаційне пошкодження, і тому їх необхідно буде замінити. Таким чином, фаза модернізації має вирішальне значення не тільки для повного використання фізичного потенціалу БАК, але й для забезпечення роботи колайдера після 2025 року.

HL-LHC буде спиратися на ряд ключових інноваційних технологій, включаючи найсучасніші надпровідні магніти з магнітним полем в 11-12 Тесла, компактні надпровідні крабові порожнини з ультра-точним контролем фази для повороту пучка [24], нову технологію для колімації пучка, надпровідні ланки з великою потужністю та ланки без втрат. Докладний опис проекту та його технологічні та експлуатаційні проблеми наведено у звіті з технічного проектування HL-LHC [19].

1.2 Експерименти БАК: CMS, ATLAS, ALICE, LHCb

У семи експериментах на великому адронному колайдері (БАК) використовуються детектори для аналізу безлічі частинок, що утворюються при зіткненні в прискорювачі. Ці експерименти проводяться за допомогою співпраці вчених з дослідницьких інститутів з усього світу. Кожен експеримент є характеризується його детекторами, що здатні дослідити якомога більший діапазон фізики.

1.2.1 Експеримент БАК CMS

Назва колаборації CMS пов'язана з багатофункціональним детектором, компактним мюонним соленоїдом (CMS) [31], що призначений для спостереження за різноманітними явищами при зіткненнях протон-протонів

та важких іонів. Його мета полягає в дослідженні природи порушення електрослабкої симетрії, що пояснюється в Стандартній Моделі завдяки механізму Хіггса [45].

Детектор CMS (рисунок 1.2) є багатоцільовим детектором, призначеним для вивчення Стандартної моделі елементарних частинок та вивчення нових фізичних властивостей за її межами. Відкриття на детекторі CMS бозону Хіггса, що було підтверджено експериментом ATLAS, є одним з найбільших успіхів співпраці CMS та ATLAS.

Експеримент CMS запланував значні оновлення детектора для досягнення цілей програми на HL-LHC. Крім того, експеримент CMS розраховує зібрати щонайменше в 150 разів більше даних до кінця 2030 в порівнянні з БАК Run1, що проходив в 2009 – 2013 рр.

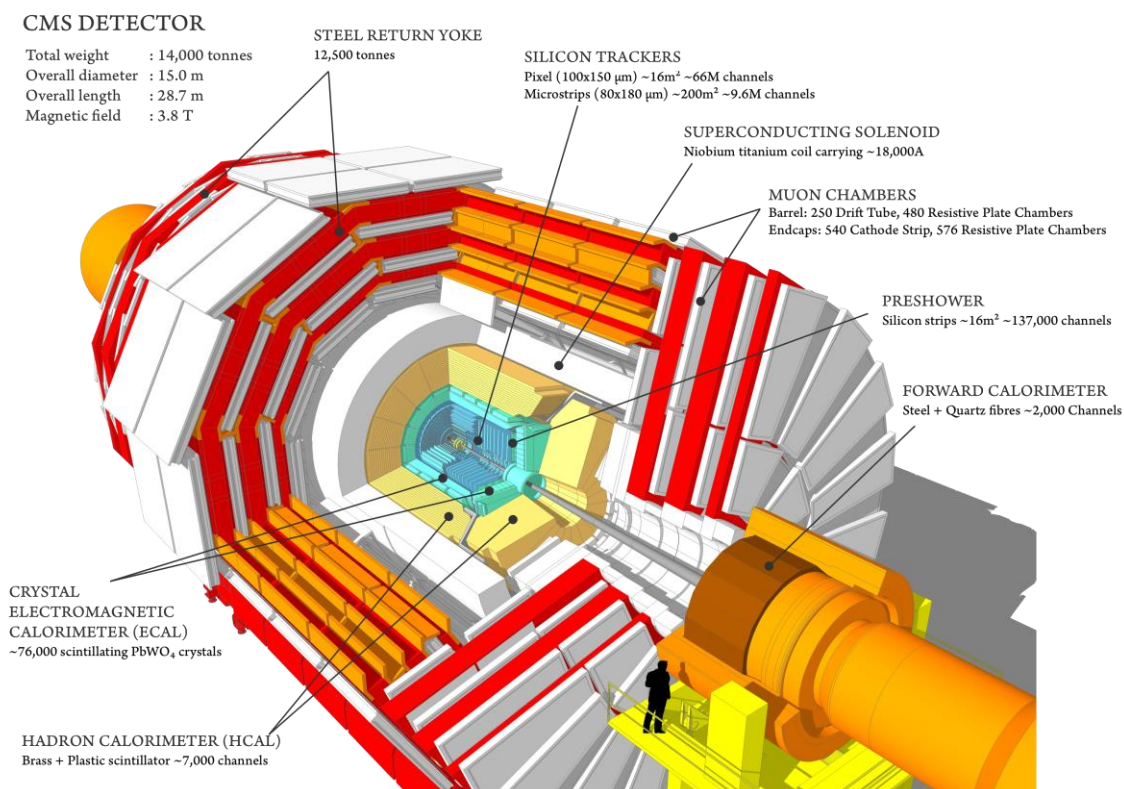


Рисунок 1.2 – Експеримент CMS та його детектор

1.2.2 Експеримент БАК ATLAS

Основною метою експерименту ATLAS (A Toroidal LHC ApparatuS) (рисунок 1.3) [1] є дослідження нової фізики поза Стандартною моделлю елементарних частинок, що виникає при надзвичайно високих енергіях у БАК. Він також шукає існування темної матерії і додаткових вимірів. Детектор виконаний з чотирьох основних шарів: магнітна система, що згинає траєкторії заряджених частинок; внутрішній детектор, який вимірює шлях заряджених частинок; калориметри, що ідентифікують фотони, електрони і струмені; мюонний спектрометр, який розпізнає присутність мюонів. Апарат має наступні розміри: довжину 46 м, діаметр приблизно 25 м та вагу приблизно 70000 тонн.

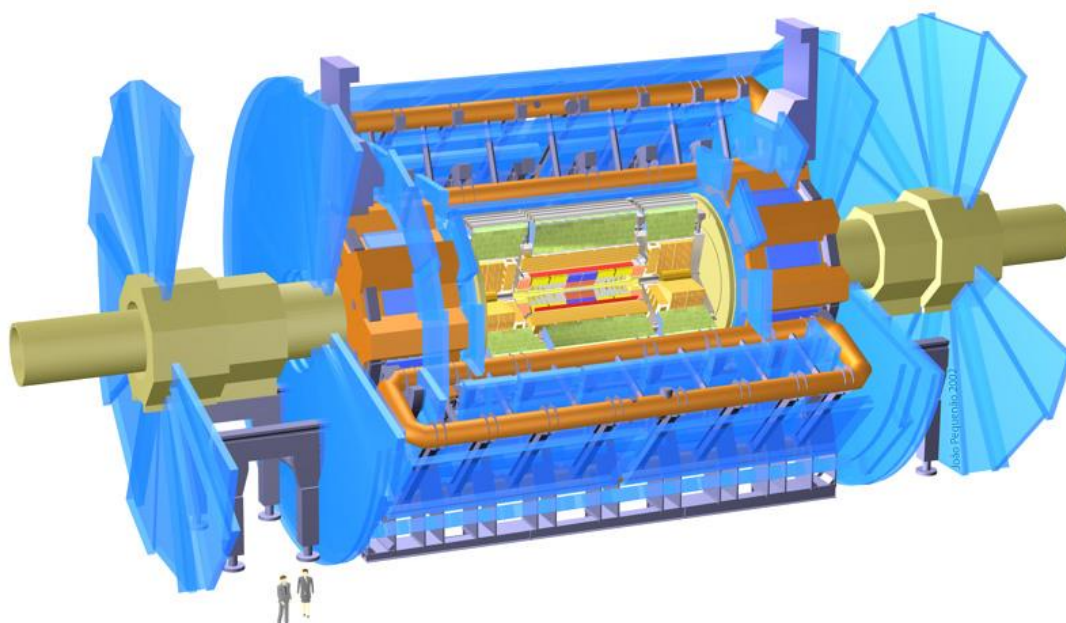


Рисунок 1.3 – Детектор експерименту ATLAS

1.2.3 Експеримент ВАК ALICE

Експериментальна колаборація ALICE [2] (рисунок 1.4) має універсальний детектор для дослідження зіткнень важких важких ядер (Pb) при високих енергіях, вивчає сильну взаємодію, зокрема кварк-глюонну плазму при екстремальних значеннях щільності енергії та температури. Цей детектор призначений для виявлення великої кількості окремих подій, які відбуваються при кожному зіткненні важких ядер. Детектор важить 100000 тонн і складається з бочкоподібної частини детектора, яка вимірює адрони, електрони і фотони, і мюонного спектрометра в передній області.

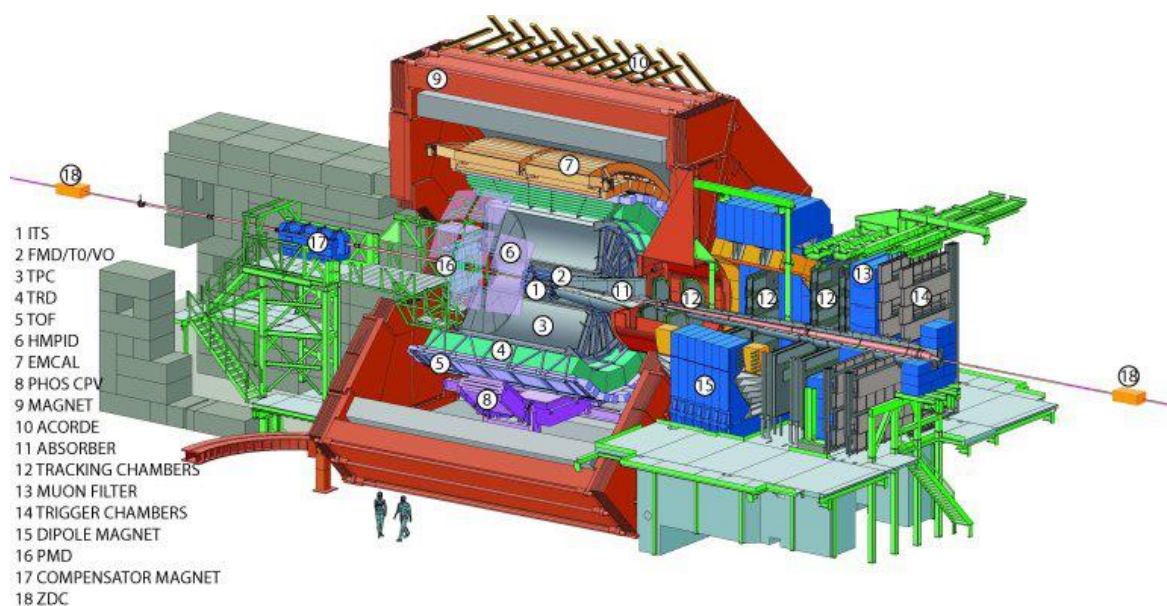


Рисунок 1.4 – Детектор експерименту ALICE

1.2.4 Експеримент ВАК LHCb

Детектор в експерименті LHCb (Large Hadron Collider for beauty) [11] спеціалізується на дослідженні В-мезону. Зокрема, у цьому експерименті вивчається декілька аспектів фізики важких іонів, електрослабкої взаємодії та

квантової хромодинаміки. Детектор LHCb складається з двох типів детекторів: системи спостереження та системи ідентифікації частинок, які разом реконструюють подію.

1.3 Інші не-ВАН експерименти

Новий експеримент SHiP («Пошук прихованих частинок») [7] (рисунк 1.5) призначений для пошуку слабо взаємодіючих важких частинок з масою в декілька гіга електронвольт. Оскільки інтенсивний тау-нейтринний потік виробляється шляхом розпаду дивного Ds-мезону, експериментальна установка включає нейтринний детектор для дослідження поперечного перерізу реакцій народження та виявлення тау-нейтрино. Цей детектор також підходить для виявлення частинок темної матерії через розсіювання на них електронів атомів мішені.

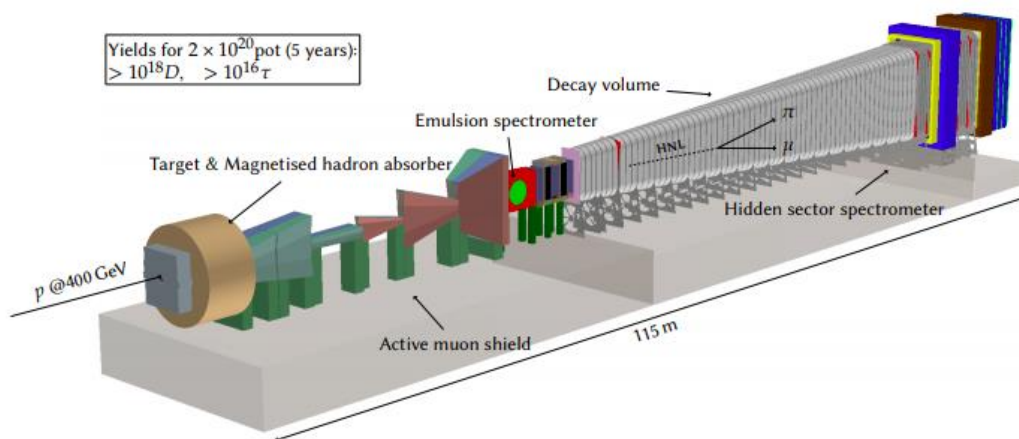


Рисунок 1.5 – Детектор експерименту SHiP

1.4 Обчислювальні ресурси, що використовуються для аналізу даних ВАН

ЦЕРН має одне з найвимогливіших обчислювальних середовищ у

науковому світі. Відмітимо, що інтернет-мережа була придумана та розроблена в ЦЕРНі для автоматизованого обміну інформацією між вченими в університетах та наукових інститутах по всьому світу. Починаючи від розробки програмного забезпечення для обробки та зберігання даних, комп'ютерної підтримки мереж та експериментальних програм ВАК та інших програм, автоматизації та управління експериментами, до надання різних послуг та сервісів для комплексу прискорювачів та для всіх лабораторій ЦЕРН та їх користувачів, обчислювальні ресурси лежать в основі всієї інфраструктури в ЦЕРНі (рисунок 1.6).

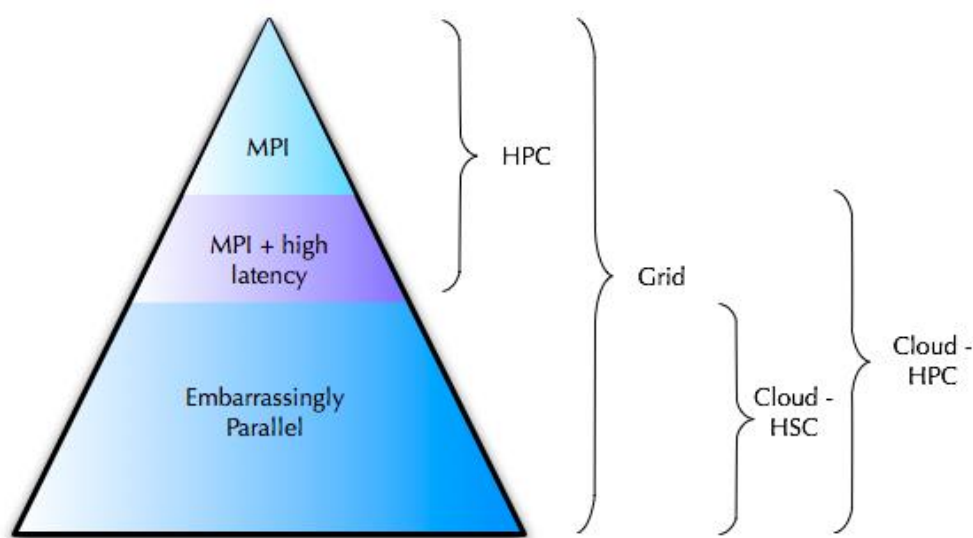


Рисунок 1.6 – Піраміда використання ресурсів в не-гетерогенному обчислювальному середовищі, наприклад в ЦЕРНі

1.4.1 Грід-обчислення

Обчислювальна грід мережа WLCG (Всесвітня обчислювальна грід мережа) [25] складається з чотирьох рівнів, які ідентифікують як Рівень 0, Рівень 1, Рівень 2 та Рівень 3. Кожен рівень складається з декількох

комп'ютерних центрів і надає певний набір послуг. Між ними знаходяться яруси «tiers», що обробляють, зберігають та аналізують всі дані з Великого Адронного Колайдера (ВАК).

Рівень 0 – це Центр обробки даних ЦЕРН. Всі дані з ВАК проходять через цей центральний вузол, але він забезпечує менше 20% загальної обчислювальної потужності ґрид. ЦЕРН відповідає за безпечне зберігання необроблених даних (мільйони цифрових показань з усіх детекторів) і виконує перший перехід при реконструкції вихідних даних у змістовну інформацію. Рівень 0 розподіляє необроблені дані та передає їх до рівня 1 або повторно обробляє дані, коли ВАК не працює.

Рівень 1 складається з 13-ти комп'ютерних центрів, достатньо великих для зберігання даних ВАК. Вони забезпечують цілодобову підтримку ґрид-інфраструктури і несуть відповідальність за збереження пропорційної частки необроблених і реконструйованих даних, а також за здійснення великомасштабної обробки і зберігання відповідних результатів, поширення даних до центрів Рівня 2 і зберігання малої частки даних, які виробляють центри Рівня 2. Оптико-волоконні ланки, що працюють на швидкості 10 гігабіт в секунду, підключають ЦЕРН до кожного з 13-ти основних центрів Рівня 1 по всьому світу. Ця виділена високошвидкісна мережа називається LHC Optical Private Network (LHCOPN) [82].

1.4.2 Хмарні обчислення

Три провідні європейські дослідницькі центри об'єдналися з комерційними постачальниками послуг, щоб створити європейську платформу для хмарних обчислень, яка може забезпечити зберігання та обробку даних для декількох великих науково-дослідних проєктів в Європі, починаючи з Великого Адронного Колайдера в ЦЕРН.

За підтримки ЦЕРН, Європейської лабораторії молекулярної біології

(EMBL) [51], Європейського космічного агентства (ESA) і провідних ІТ-провайдерів, наукова хмара Helix Nebula підтримуватиме масові ІТ-вимоги європейських вчених і стане доступною для державних організацій і промисловості після початкової пілотної фази. Протягом двох років EMBL та ESA використовуватимуть наукову хмару для дослідницьких проектів та надання її обчислювальної інфраструктури за вимогою вчених цих організацій.

1.4.3 Волонтерські ресурси

BOINC (Berkeley Open Infrastructure для мережевих обчислень) – це програмна платформа з відкритим вихідним кодом для розподілених обчислень, в якій задіяні волонтерські обчислювані ресурси. Вона використовує бездіяльні цикли процесорів учасників програми для виконання наукових розрахунків [17].

Служба BOINC у ЦЕРН розміщує сервери BOINC для LHC@home [78]. Його перше застосування – це SixTrack [101], який імітує частинки, що циркулюють навколо кільця БАК, для вивчення довгострокової стабільності орбіт частинок. Другий додаток – Test4Theory [77], який використовує технологію віртуальних машин для запуску моделювання подій БАК на хост-комп'ютерах. Додатковими споживачами є використання системи для моделювання експерименту ATLAS та нещодавно також для моделювання експериментів LHCb і CMS.

1.5 Поточний стан оптимізації ефективності обчислень у ФВЕ

За своїм розміром і складністю дані, згенеровані та оброблені сучасними експериментами ФВЕ, зайняли місце того, що відомі як Великі

Дані (Big Data) [81] (рисунки 1.7 – 1.11). Наприклад, експеримент CMS виробив, реконструював, зберіг, передав та аналізував більше 10 мільярдів змодельованих подій під час Run 1 у 2009-2013 роках. Обсяг даних, зібраних та збережених експериментами ВАК до 2013 року, склав близько 15 PB на рік, не так далеко від 180 PB на рік, завантажених у Facebook, 98 PB даних у пошуковому індексі Google або 15 PB на рік відео, завантаженого в YouTube. До кінця 2016 року кількість даних збільшилася до 50 PB на рік.

З точки зору програмного забезпечення, засоби та алгоритми для моделювання та реконструкції є досить дорогими, як у плані ресурсів для розробки, як і для обчислюваних ресурсів. Спільнота ФВЕ витратила близько 100 мільйонів доларів США тільки на розробку та підтримку програмного забезпечення для моделювання та реконструкції подій в експериментів на ВАК. Крім того, для експерименту CMS частка витрат на апаратний життєвий цикл, що використовується для цілей моделювання, його обслуговування та експлуатацію становить близько 5-10 мільйонів доларів США на рік.

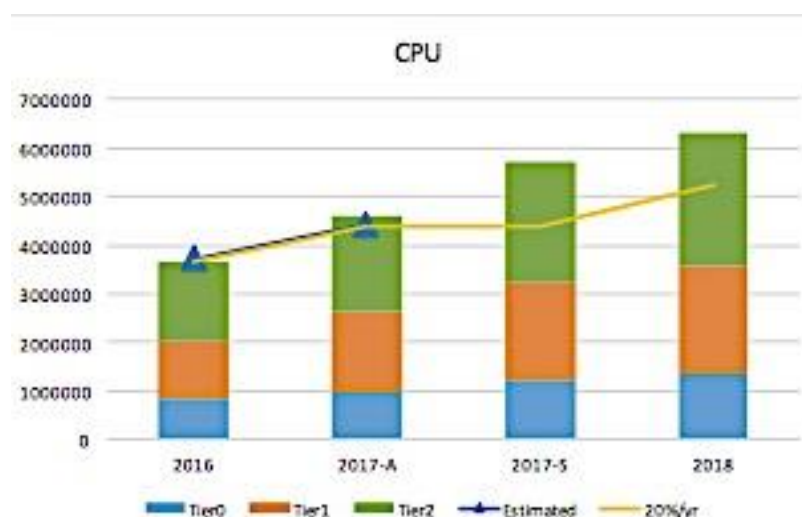


Рисунок 1.7 – Кількість центральних процесорних (ЦП) ресурсів, що заплановані для використання в HL-LHC

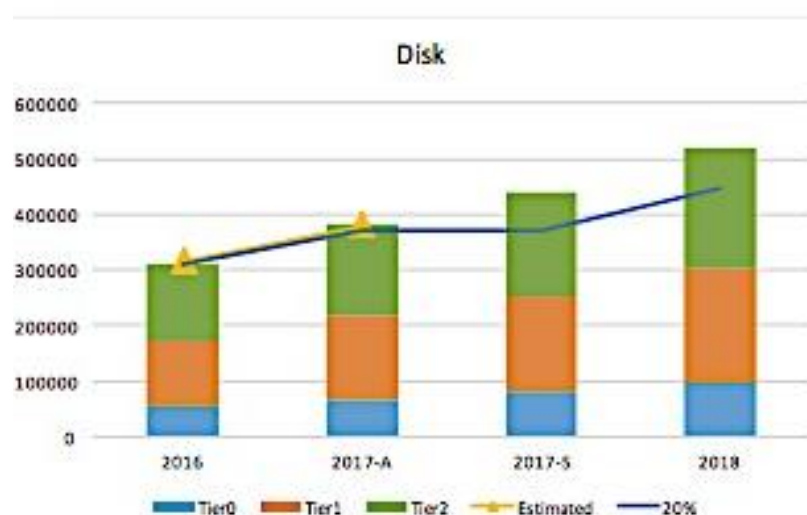


Рисунок 1.8 – Дисккові ресурси, що заплановані для використання в HL-LHC (у Терабайтах)

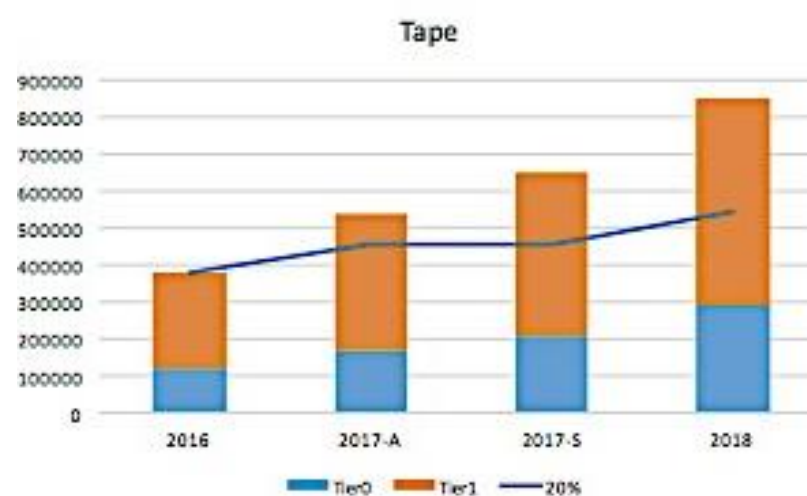


Рисунок 1.9 – Стрічкові ресурси, що заплановані для використання в HL-LHC (у Терабайтах)

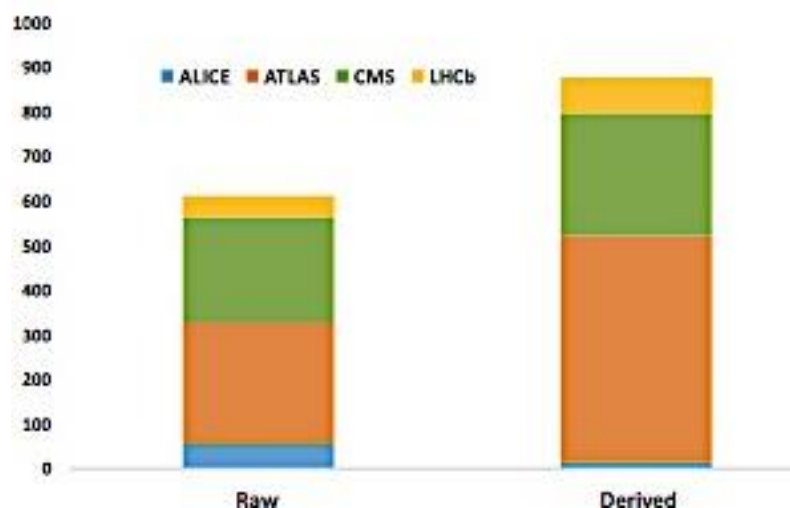


Рисунок 1.10 – Кількість даних, що планується зібрати та заплановано для використання в аналізі для HL-LHC (у Петабайтах)

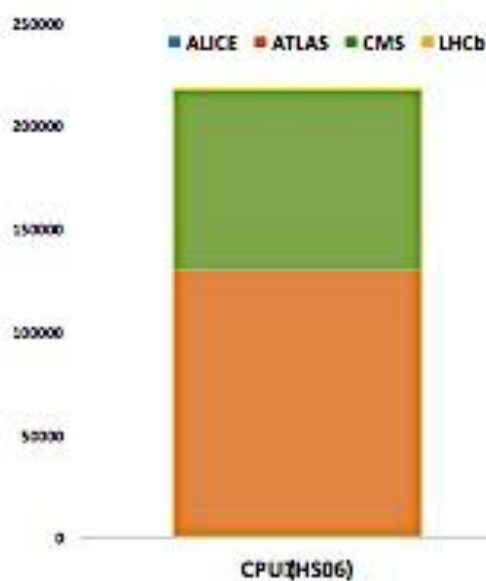


Рисунок 1.11 – ЦПУ ресурси, які планується використовувати в HL-LHC (у HS06 вимірюваннях)

1.5.1 Оцінка ефективності ресурсів

У роботі [30] було проаналізовано процес організації комп'ютингу в грид, хмарних та волонтерських обчисленнях та наскільки в них

використовується різні типи процесорів, з великою кількістю різних конфігурацій, де кожен обчислювальний слот має свою власну обчислювальну здатність.

Цей аналіз вказує на необхідність проведення тестування обчислювальних ресурсів, які будуть залучені для виконання тих чи інших завдань. Для цього використовується так зване «пілотне тестування». Робота «пілота» або агента починається з обчислення продуктивності для наявного обчислювального слота, а саме перед запитом на роботу, потім «пілот» або агент тестує обчислювальні ресурси, щоб переконатися, що обчислювальний слот може бути доступним до кінця виконання завдання. Процедура тестування робить використання комп'ютерних ресурсів більш ефективним.

Як приклад для критерію ефективності використання ресурсів застосовується фітнес-функції вимірювання «потужності роботи»:

$$\frac{CPU\ Time}{Events}. \quad (1.1)$$

В цьому випадку, оскільки основні інтенсивні фази роботи процесора в моделюванні транспорту подій задіяні у самому моделюванні та реконструкції подій, зазвичай для критерію ефективності найбільш відповідним є лінійна залежність процесорного часу (CPU Time) від кількості подій (Events).

Отримання такої залежності можна розглядати як застосування практики порівняльного аналізу, а саме використання статичного підходу, який може дати переваги, але не зможе допомогти в налаштуванні та оптимізації продуктивності у неоднорідному середовищі.

1.6 Опис CAE-системи моделювання транспорту частинок GeantV

Програмний пакет GeantV є сучасним розвитком CAE-системи автоматизованого проектування Geant4 [5, 9, 65]. Geant4 моделює транспорт частинок, що проходять через матерію та взаємодіють з нею. Прикладними галузями Geant4 є фізика високих енергій, ядерна фізика та фізика прискорювачів, а також дослідження в галузі космосу, матеріалознавства, медицини, біології, безпеки та має застосування у промисловості.

У ФВЕ використання додатків Geant4 для моделювання є незамінним при проектуванні детекторів, розробці та оптимізації алгоритмів реконструкції треків частинок, а також моделюванні сигналу і фону для подій, що пов'язані з фактичним аналізом даних.

Моделювання із застосуванням CAE-системи автоматизованого проектування Geant4 є дуже складними і одними з наймісткіших обчислювальних завдань для обробки експериментальних даних у ФВЕ. Так, протягом роботи БАК, за весь етап Run-I (2011 – 2013), більше ніж половина ресурсів Worldwide LHC Computing Grid (WLCG) була використана для симуляції 10^{10} подій (еквівалент часу 10^{12} секунд процесорного часу).

Симуляції Geant4 містять велику кількість інформації із сучасної фізики високих енергій та ядерної фізики, від теорії до фізики експерименту. Як правило, його застосування потребує великих обсягів обчислювального часу та пам'яті, в основному через складність обчислювань для складної геометрії детекторів, через які транспортуються частинки. Саме ці проблеми ускладнюють пошук нових концепцій для обробки експериментальних даних та створення програмних інструментів, які можуть системно аналізувати дані, створюючи несуперечливі та, що найважливіше, відтворювані результати.

Проект GeantV [13] має увібрати в себе та розвинути всі позитивні сторони CAE-системи Geant4 [5]. Він розпочався у 2013 р. з проведення R&D

за декількома напрямками, зокрема такими: збільшення частки паралелізму і пропускної спроможності за допомогою включення SIMD (single instruction, multiple data – одиночний потік команд, множинний потік даних) в код та покращення локальності застосування, щоб уникнути непотрібних навантажень на комп'ютерну архітектуру та зайвих процесів. Це уможливило ефективне використання САЕ-системи GeantV на багатоядерних векторних архітектурах таких, як Intel KNL [110], і оптимізувати використання кеш-пам'яті системи.

Метою проекту GeantV є оптимізація алгоритмів моделювання аби отримати максимальну ефективність від використання розподілених масивів для паралельних архітектур, через знаходження оптимального значення параметрів, що визначають обчислювальну продуктивність, а саме: кількості флопс (операцій з рухомою комою), розміру зовнішньої пам'яті з високою пропускною здатністю, використання ієрархій пам'яті та кеш-пам'яті тощо. Оскільки на виході необхідно оптимізувати велику кількість параметрів що пов'язані з складною неоднорідною комп'ютерною інфраструктурою для паралельних обчислень, завдання оптимізації продуктивності GeantV доводиться розглядати, як оптимізацію проблеми «чорного ящика».

GeantV (це стосується і Geant4), як САЕ-система автоматизованого проектування має різноманітні можливості для обробки даних з геометрії детектора експерименту, відстеження треків частинок, відповіді детектора, керування запуском задач, візуалізацією та графічним інтерфейсом користувача. Використання цих під-пакетів допомагає дослідникам витратити менше часу на процеси налаштування системи для обробки даних на технічному рівні та зосередитись на важливих аспектах моделювання.

Нижче коротко описані важливі модулі, що представлені в програмовому пакеті GeantV:

- модуль для моделювання геометрії експерименту виконує аналіз налаштування експерименту, включаючи детектори та поглиначі, з

урахуванням того, як геометрія детектору вплине на траєкторію частинок в експерименті;

- модуль для моделювання відстеження треків частинок виконує моделювання проходження частинки через речовину (це передбачає розгляд можливих взаємодій частинки з елементами детектора та процесів розпаду частинок);

- модуль для моделювання відповіді детектору виконує запис інформації про проходження частинки через об'єм детектора та процесу реакції детектору на цей процес;

- модуль для керування запусками завдань виконує детальний запис результатів симуляції набору подій (циклу), а також налаштування симуляції для різних конфігурацій між циклами задач;

- GeantV також пропонує ряд модулів для візуалізації, включаючи бібліотеку OpenGL [118].

Оцінити продуктивність такої складної САЕ-системи з врахуванням використаної пам'яті, часу, кількості інструкцій або розміру використаного кешу дуже важко через брак явного виду функцій, або їх чисельного представлення.

Моделювання за допомогою принципу «чорного ящика» зазвичай використовується у випадках, коли в процесі обробки даних постає задача підгонки даних, залежність яких складно визначити або які не можуть бути описані конкретно математичною моделлю. Також цей підхід може використовуватися у випадках, коли замість властивостей і взаємозв'язків складових частин системи необхідно вивчати реакцію цілої системи на зміни. Для цього підходу існує набір інструментів, які пропонують декілька лінійних та нелінійних моделей для проблеми «чорного ящика», що традиційно використовується для представлення динамічних систем. Структурні елементи цих моделей різняться залежно від гнучкості інструментів, необхідних для оцінки динаміки та шуму в системі.

Моделювання «чорного ящика» зазвичай є процедурою пробного оцінювання результатів задачі, де моделюються параметри задачі та порівнюються результати задачі при зміні цих параметрів. Як правило, дослідження починаються з простої структури лінійної моделі та вже потім ускладнюється для складніших моделей.

Ефективність стратегій для оптимізації задачі «чорного ящика» зазвичай вимірюється кількістю проведених оцінок для досягнення оптимального значення або навпаки з погляду еволюції величини оцінки наближення до оптимального значення цільової функції для певної кількості оцінок.

Еволюційні алгоритми пропонують оптимізувати цільову функцію за допомогою процесу штучної еволюції для вибору найкращих параметрів. Наприклад, в генетичних алгоритмах значення параметрів розглядають як особини в поколіннях, які розвиваються від покоління до покоління в середовищі, що уможливлює виживання та відтворювання найпридатніших особин (найбільш придатними є найкращі параметри для цільової функції).

1.7 Теорія та алгоритми, що використовуються в дослідженні

В задачах багатокритеріальної оптимізації в умовах реального часу виникає багато складнощів, включаючи наявність шуму в оцінці функцій, складність розрізнювання глобального оптимального розв'язку від локальних оптимальних розв'язків, багатовимірність розв'язку, труднощі, що пов'язані з нетривіальними обмеженнями в задачі, та відсутність стаціонарності в розв'язку через те, що умови задачі можуть змінюватися з часом [41].

Пошук найкращого розв'язку багатокритеріальної задачі фактично означає пошук екстремальних значень для набору функцій, які залежать від багатьох змінних. Цей набір функцій часто називають «функціями

пристосованості» або «цільовими функціями» (аналог у фізиці – гамільтоніан системи).

Стохастична оптимізація є одним з основних підходів в обчислювальній статистиці. Стохастична оптимізація означає мінімізацію (або максимізацію) цільових функцій за наявності випадковості в процесі оптимізації. Випадковість може бути пов'язана як із шумом у вимірюваннях, так із застосуванням методу Монте-Карло в процедурі моделювання, або ж з обома випадками одночасно. Стохастичні методи оптимізації призначено для оцінки функцій за наявності шуму [83].

Важливо зазначити, що, навіть коли набори даних згенеровано за допомогою точних вимірювань, для прискорення знаходження розв'язку задачі оптимізації існують деякі методи, які додають випадковості у процес пошуку. Процес випадковості може зробити метод менш чутливим до помилок моделювання, а також дати змогу методу вийти з локального оптимуму і зрештою підійти до глобального оптимуму. Відомі способи стохастичного пошуку – це алгоритм імітації відпалу [72], крос-ентропійні методи [43], стохастичне тунелювання [116], стохастичний градієнтний спуск [115], метод скупчення частинок [120], а також еволюційні алгоритми: генетичні алгоритми [49] та еволюційні стратегії [57].

Алгоритми стохастичного пошуку – це методи, пов'язані з процедурою вибору для розв'язання багатьох складних комбінаторних задач [61]. Основним мотивом нинішнього інтенсивного розвитку алгоритмів стохастичного пошуку є:

- використання алгоритмів, що виникають в природничих науках;
- впровадження нових схем стохастизації;
- використання гібридних та змішаних стратегій пошуку;
- підвищення гнучкості і надійності алгоритму;
- можливість використання складних структур даних та вдосконалених алгоритмічних підходів для новітніх апаратних архітектур та засобів

ефективного оброблювання даних.

Серед алгоритмів стохастичного пошуку є вкрай цікаві алгоритми, що використовують методи стохастичної оптимізації локального пошуку. Вони ефективні для визначення пошукового простору та цілей і включають спектр методів, починаючи від простої процедури конструювання та ітерацій до складніших методів.

Обчислювальні підходи до розв'язку складних комбінаторних задач можна розглядати як такі, що виконують певну форму пошуку. По суті, пошукові алгоритми генерують та оцінюють варіанти кандидатів розв'язку для конкретної задачі в режимі реального часу. Для комбінаторних розв'язків задачі оцінка кандидата для конкретного розв'язку вимагає перевірки, чи дійсно розв'язок кандидата може задовольнити всі зазначені обмеження. Для комбінаторних задач оптимізації він передбачає обчислення значення заданої цільової функції [62].

Для комбінаторних задач застосовують процедури групування, впорядкування або вибору окремих наборів об'єктів, але вони, як правило, мають високу обчислювальну складність. Комбінаторні задачі належать до класу NP-складних задач (NP-складна задача – це клас задач, що є принаймні такими ж складними, як і найскладніші задачі в NP, а саме – недетерміновані поліноміальні задачі). Винятком є такі задачі, як мінімальні кістякові дерева або метод найкоротшого шляху, що його використовують у багатьох галузях комп'ютерних наук, інженерії, хімії, біології та фізики.

Більшість алгоритмів стохастичного пошуку на практиці добре працюють, але теоретично мало проаналізовані, і не гарантують знаходження оптимальних розв'язків. Ми можемо використовувати емпіричні методи для аналізу та характеристики алгоритмів стохастичного пошуку. Найцікавішими аспектами стохастичного пошуку є перевірки результатів на мінливість, що пов'язана з рандомізацією, стійкістю різних типів задач та їхніх параметрів, а також масштабованістю, пов'язаною з розмірністю задачі.

Налаштування алгоритмів стохастичного пошуку значною мірою залежить від відповідної параметризації та налаштування всієї системи пошуку. Треба зазначити, що в цьому випадку налаштування може бути дуже трудомістким через відсутність повного розуміння, наскільки продуктивність додатку залежить від налаштувань параметрів. Ще однією перевагою алгоритмів стохастичного пошуку є зручна паралізація, особливо для еволюційних алгоритмів (як для дрібноблочних, так і для багатоблочних паралельних підходів до обчислювань).

Таким чином, якщо класичні методи детермінованої оптимізації (лінійне та нелінійне програмування) є ефективними для конкретного ряду задач, то стохастичні методи дають змогу розв'язувати багато задач, для яких детерміністичні методи не працюють або ж працюють неефективно.

1.7.1 Задача багатокритеріальної оптимізації

Перший крок полягає у визначенні багатокритеріальної оптимізації (БКО) та її завдань. БКО називають також багатоцільовою або векторною оптимізацією. Головними елементами БКО є вектори розв'язків, обмеження, цільовий простір, простір розв'язків та функції пристосованості:

– вектори розв'язків (набір n прийнятних розв'язків або вектор вимірювань)

$$\mathbf{x} = [x_1, x_2, \dots, x_n]. \quad (1.2)$$

– обмеження описують залежності між компонентами вектора розв'язків та можуть бути задані рівностями і нерівностями

$$\mathbf{g}_i(\mathbf{x}) \leq 0 \quad \text{or} \quad \mathbf{h}_i(\mathbf{x}) = 0. \quad (1.3)$$

– простір можливих розв’язків є підмножина прийятних розв’язків $X \subset R^n$ (рисунок 1.12).

– цільовий простір $Y \subset R^q$ – це множина відповідних прийятних значень цільової функції $\vec{y} = \vec{f}(\vec{x})$, $\vec{x} \in X$ (1. 12).

– функція f , в різних випадках, називається цільовою функцією, функцією вартості (мінімізації), функцією корисності або функційності (максимізації), або ж функцією енергії чи енергією. Розв’язок, який мінімізує (або максимізує, якщо це на меті) цільову функцію, називається оптимальним розв’язком.

Головні особливості БКО представлено на рисунку 1.12.

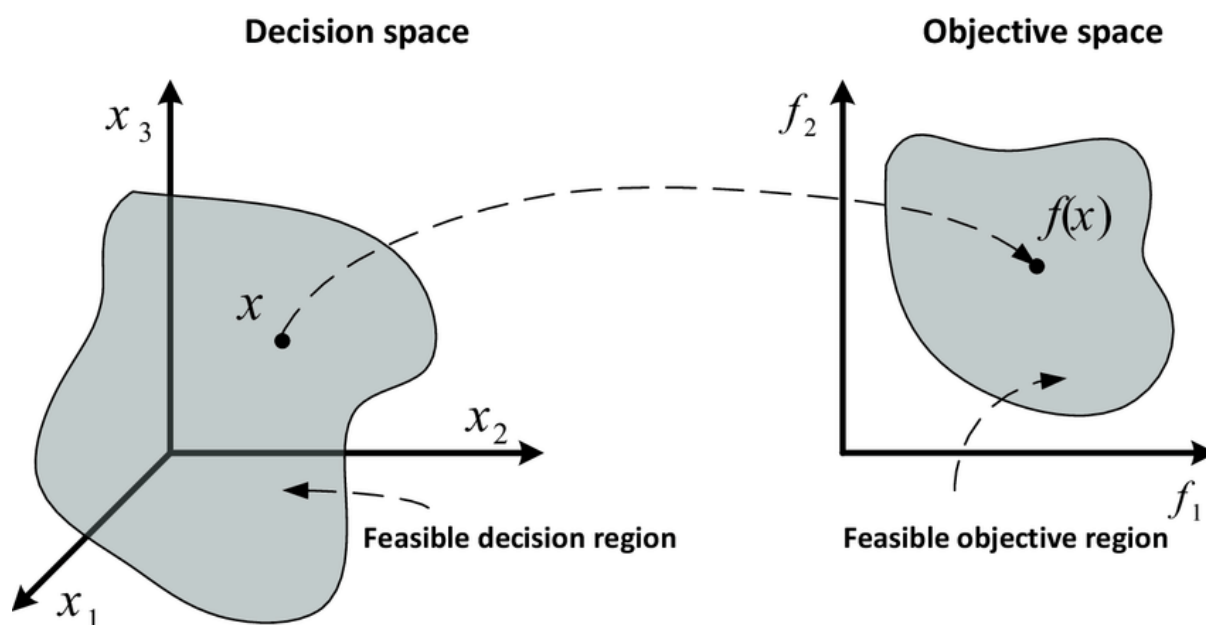


Рисунок 1.12 – Простір розв’язків та об’єктивний простір для задачі БЦО

1.7.2 Визначення оптимуму Парето

Френсіс Ю. Еджворт (1845-1926) та Вільфредо Парето (1848-1923) вважаються першими, хто представили концепцію неповноцінності в задачах багатокритеріальної оптимізації в контексті економіки. Від того часу багатокритеріальна оптимізація була впроваджена в

інженерно-конструкторські галузі і розвивалася швидкими темпами в інших галузях. Важливість границь Парето виходить далеко за рамки визначення «найкращого дизайну» або «домінантних» розв'язків і полягає в аналізуванні технологічної інфузії в існуючих системах, виборі архітектури системи та інженерних систем з життєвим циклом ре-конфігурування [93].

Окремий розв'язок для задачі багатокритеріальної оптимізації вважається оптимальним за Парето (анг. Pareto-optimal) або ефективним за Парето (анг. Pareto-efficient), якщо жоден інший варіант розв'язку задачі не може покращитись за однією ціллю, одночасно не погіршуючи принаймні одну з інших цілей [29].

Набір розв'язків для задачі багатокритеріальної оптимізації, відомий як ефективний набір або множина Парето, містить всі можливі розв'язки задачі, оптимальні за [113].

Методи, розвинуті Парето, спрямовано на знаходження набору ефективних розв'язків, причому цільові вектори, які відповідають цим розв'язкам, є недомінантними в N -вимірному об'єктивному просторі.

Розглянемо цільовий простір $Y \subset R^q$. Відношення часткового порядку $<$ відображає предметне домінування або Парето-оптимальною преференцією і визначається у цільовому просторі наступним чином

$$\vec{y} < \vec{y}' \Leftrightarrow \forall i \quad (1 \leq i \leq q), \quad y_i \leq y'_i \wedge \exists j \quad y_j < y'_j, \quad (1.4a)$$

$$\vec{x} < \vec{x}' \quad \text{якщо} \quad \vec{y} = \vec{f}(\vec{x}) < \vec{y}' = \vec{f}(\vec{x}'), \quad (1.4b)$$

де y_i визначає i -ту компоненту вектора \vec{y} та y_j визначає j -ту компоненту вектора \vec{y} . Співвідношення (1.4b) визначає Парето-оптимальне рішення у просторі розв'язків.

Аналогічно визначимо відношення переваги \preceq , яке називається

домінуванням або рівністю

$$\forall \vec{y}, \vec{y}' \in R^q \quad \vec{y} \preceq \vec{y}' \quad \Leftrightarrow \quad \vec{y} < \vec{y}' \quad \vee \quad \vec{y} = \vec{y}'. \quad (1.5)$$

Задану множину векторів Y ми позначимо $\mathcal{N}(Y)$ як множину недомінантних векторів в Y наступним чином

$$\mathcal{N}(Y) = \{\vec{y}^{(*)} \in Y \mid \vec{y} \in Y \quad \vec{y} < \vec{y}^{(*)}\}. \quad (1.6)$$

Позначимо $\alpha_i = \min_{\vec{y} \in \mathcal{N}(Y)} \{y_i\}$ та $\beta_i = \max_{\vec{y} \in \mathcal{N}(Y)} \{y_i\}$. Множина $\mathcal{N}(Y)$ є обмеженою ідеальною точкою $\vec{\alpha} = (\alpha_1 \dots \alpha_q)$ та анти-ідеальною точкою або точкою надира $\vec{\beta} = (\beta_1 \dots \beta_q)$. Ідеальну точку можна обчислити, оптимізуючи кожну цільову функцію окремо. Проте для $q > 2$ важко обчислити точку надира, не обчисливши всієї множини недомінантних розв'язків.

1.8 Генетичні алгоритми як спеціальний випадок алгоритмів стохастичної оптимізації

Генетичні алгоритми (ГА) [33] (рисунок 1.13) являють собою популярний підхід до стохастичної оптимізації, як метод розв'язку глобальної задачі оптимізації через пошук найкращого розв'язку серед декількох локальних мінімумів. Стохастичну оптимізацію визначають як знаходження глобального мінімуму або максимуму сукупності цільових функцій за наявності випадковості в процесі оптимізації. Генетичний алгоритм базується на принципах природного відбору і є спеціальним випадком загальних класів еволюційних алгоритмів, де окремі особини в генетичному алгоритмі можуть мати дійсний, дискретний або комплексний вигляд.

Для дослідження оптимізації роботи САЕ-системи автоматизованого проектування GeantV ми застосовуємо генетичні алгоритми, методи машинного навчання без вчителя, включаючи багатовимірний статистичний аналіз [16]. Ці підходи придатні для розв'язування багатокритеріальної задачі «чорного ящика» з великими об'ємами обчислень для оцінки функції пристосованості. Гарним прикладом у цьому випадку є генетичні алгоритми з недомінантним сортуванням (NSGA-II [42] або NSGA-III [36, 66] алгоритми). Головним недоліком схеми генетичних алгоритмів є відсутність додаткових процедур перевірки або спеціальних операторів, які створюють механізми прискорення збіжності цих алгоритмів до множини глобальних оптимальних точок.

Фундаментальна відмінність ГА від досліджень за допомогою випадкових алгоритмів або алгоритмів імітації відпалу, полягає в тому що, генетичний алгоритм працює з популяцією кандидатів для конкретного розв'язку задачі. Генетичний алгоритм одночасно розглядає декілька варіантів розв'язування задачі мінімізації функції втрат через ітерації, переміщаючи популяцію розв'язків до глобального оптимуму. Зокрема, генетичні алгоритми, як правило, є найпристосованішими для задач, де оцінювання функцій втрат здійснюється за допомогою комп'ютерних обчислень, наприклад таких, як числове моделювання [73].

Як відомо, біологічні системи є найбільш оптимізованими, адаптивними системами. Вважається, що еволюція створює оптимальні, адаптивні системи з використанням еволюційних принципів для обчислення оптимальних розв'язків [59]. Головною ідеєю розвинення такого підходу був механізм створення адаптивних систем штучного інтелекту. Зауважимо, що сучасне використання обмежується ідеєю використання алгоритмів для розв'язання комбінаторних задач оптимізації.

Визначимо основні еволюційні (генетичні) принципи:

- виживання найпридатніших;

- гени виживають і об'єднуються шляхом спарювання;
- схрещування (кросоверинг) (закони Менделя) [80];
- мутація (випадкові зміни в генетичному матеріалі) [80];
- перестановки (випадковий генетичний механізм рекомбінації) [44];
- локальне вдосконалення особин («живильне середовище»).

Розглянемо основні параметри для налаштування генетичного алгоритму. Одним з головних параметрів є розмір популяції N . Велика популяція містить представників з більшої кількості гіперплощин в просторі особин. У цьому випадку генетичний алгоритм виконує ґрунтовніший пошук. Проте велика популяція потребує більшої кількості оцінок, що спричиняє неприйнятно низьку швидкість збіжності.

Наступним важливим параметром є частота схрещування C яка пов'язана з частотою застосування оператора схрещування \hat{C} . Більша частота призводить до швидшої генерації нових структур, але може призвести до відмови від них, перш ніж селекція може привести до покращення. Занадто низьке значення частоти може спричинити стагнацію, тоді як частота мутацій M підвищує змінність популяцій. Приблизно $M \cdot N$ мутацій відбувається на одне покоління. Це дає змогу запобігти процесам збіжності в один і той самий локальний оптимум.

Розрив (люфт) між поколіннями G – ще один з параметрів, що описує відсоток популяції, який буде оновлений в новому поколінні. Отже, $N \cdot (1 - G)$ особин збережеться в наступному поколінні.

Стратегія селекції S – це метод добирання індивідуумів для репродукції. Нижче розглянемо декілька стратегій селекції:

- стратегія чистої селекції, при використанні якої кожна структура популяції відтворюється кілька разів, пропорційно до її ефективності [54];
- стратегія елітної селекції використовується після того, як завершено процедуру чистої селекції. Найкращі структури в популяції виживають неушкодженими у наступному поколінні [75].

Фактично селекція та різні етапи в елітізмі (алгоритм, що описує розвиток популяцій, при якому найбільш пристосовані особини дають приплід з більшою ймовірністю) відбуваються після оцінювання функції пристосованості (фітнес функції) для поточної популяції хромосом. При селекції відбирають підмножину хромосом для використання її в якості батьків для наступного покоління.

Для ідентифікації генетичного алгоритму дуже зручно використовувати наступні параметри:

$$GA = (N, C, M, G, S),$$

де N – це розмір популяції;

C – частота схрещування;

M – частота мутацій;

G – розрив (люфт) між поколіннями;

S – селекція.

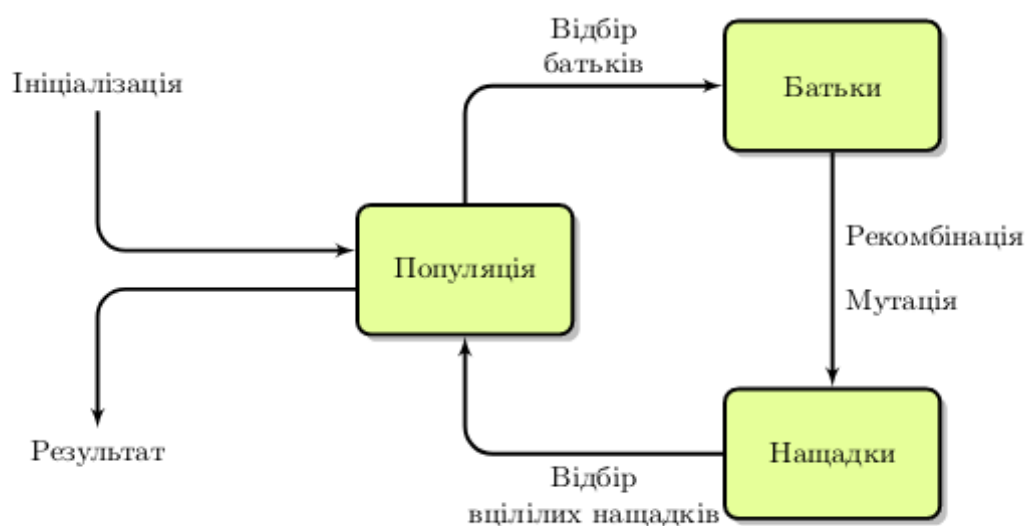


Рисунок 1.13 – Схема спрощеного генетичного алгоритму

1.8.1 Генетичний алгоритм з недомінантним сортуванням – NSGA-II/III

В останнє десятиліття одними з найбільш перспективних та широкоживаних ГА є генетичний алгоритми з недомінантним сортуванням – NSGA-II [42] та його модифікація NSGA-III [36].

Основними особливостями цих алгоритмів є:

- NSGA-II/III був розроблений для застосування ГА для оптимізації двох і більше цільових задач;
- бінарне представлення даних в алгоритмі може бути використане спільно з класичними генетичними операторами, такими як одно-точкове схрещування та точкова мутація [95];
- для неперервної оптимізації задач з декількома цільовими функціями рекомендується використовувати реалістичні представлення даних, що у свою чергу вимагає використання певних генетичних операторів таких, як імітаційний бінарний кросовер та поліноміальна мутація [34, 35].

1.8.2 Генетичний алгоритм з недомінантним сортуванням – NSGA-II

Цей генетичний алгоритм (NSGA-II) за своїми принципами роботи часто згадується як алгоритм порівняння для багатокритеріальної оптимізації. Він є дуже ефективним для визначення Парето-оптимальних рішень, що є причиною його популярності в прикладних дослідженнях. Основна робота [37, 42] була процитована 26489 разів (дані з Google Scholar на 4.01.2019), що дає нам вважати що це є одна з найуспішніших імплементацій генетичних алгоритмів.

Більш детально структуру та роботу генетичного алгоритму з недомінантним сортуванням викладено у розділі 4.

1.8.3 Генетичний алгоритм з недомінантним сортуванням – NSGA-III

NSGA-III [36, 66] використовує підхід попередньо встановленого набору контрольних точок для урізноманітнення отриманих розв’язків. У цьому підході використано нормовану гіперплощину, а саме $(M - 1)$ -мірний одиничний симплекс, який однаково нахилений до всіх цільових осей і має по одному перетину на кожну вісь. Якщо для кожної цілі розглядати p розподілень, то H M -розмірних контрольних точок (з $p + 1$ точками вздовж кожної межі) для M -цільовій задачі можна визначити як

$$H = \binom{M + p - 1}{p}. \quad (1.7)$$

NSGA-III використовує набір еталонних напрямів у просторі розв’язків для урізноманітнення одержаних розв’язків, в той час як NSGA-II задля тієї самої мети використовує адаптивну схему завдяки оператору щільності скупчення.

NSGA-III починається з вибору довільної популяції розміру N і набору широко розподілених заданих M -вимірних контрольних точок H на одиничній гіперплощині з нормалізованим вектором, перпендикулярним до цієї гіперплощини, що охоплює всю область R_+^M . Гіперплощина розміщується таким чином, щоб вона перетиналася з кожною цільовою віссю в одиниці. Розподіл популяції з N особин вибирається як найменший, кратний чотирьом, та більший за H , за умови, що для кожної контрольної точки, як очікується, буде знайдено хоча б одну особину.

Для генерації t виконуються наступні операції: по-перше, популяцію P_t сортують на різні рівні за принципом недомінантності так само, як це робиться в NSGA-II, дотримуючись принципу недомінантності. Популяцію Q_t створюють з P_t популяції, використовуючи звичайні оператори рекомбінації та мутації. Оскільки очікується, що для кожної контрольної точки буде виявлено лише одного члена популяції, то немає жодної потреби в операції селекції в NSGA-III, оскільки будь-який оператор селекції може спричинити конкуренцію між різними контрольними точками. Комбінована популяція є $R_t = Q_t + P_t$.

Далі, починаючи з першого недомінантного фронту, вибирають точки для $P_{(t+1)}$ по одній за раз доти, поки не будуть включені всі розв'язки для повного фронту. Ця процедура також ідентична процедурі NSGA-II. Останній фронт, що не може бути повністю відібраний, позначимо як F_l . Загалом, для $P_{(t+1)}$ треба вибрати лише кілька розв'язків з цього фронту, використовуючи оператор, що зберігає «нішу», який ми опишемо далі.

По-перше, кожна з популяцій $P_{(t+1)}$ та F_l нормалізується, використовуючи поточний розподіл популяції, так щоб всі цільові вектори та контрольні точки мали співвимірні значення. Після цього кожному особину з популяцій $P_{(t+1)}$ та F_l сполучають з конкретною контрольною точкою, використовуючи найкоротшу перпендикулярну відстань d від кожного члена популяції до опорної лінії, створеної шляхом з'єднання початкової точки з контрольною точкою. Потім для вибору тих точок фронту, що пов'язані з найменш представленими в $P_{(t+1)}$ контрольними точками, застосовують стратегію «виділення ніші».

Стратегія «виділення ніші» робить акцент на виборі індивіда з популяції для якомога більшої кількості виділених контрольних точок. Індивідууму з популяції, що пов'язаний з недостатньо представленою або непередставленою контрольною точкою, відразу надається перевага. Як

очікується, цей процес, відбираючи недомінантних особин, знайде одного члена популяції, що відповідає кожній даній контрольній точці, близькій до фронту Парето, за умови, що генетичні оператори рекомбінації та мутації здатні знаходити відповідні розв'язки.

Для NSGA-III офіційної програмної реалізації не існує, що затруднює порівняння різних модифікацій NSGA-III. Основна причина, чому розрізнялися всі попередні реалізації, полягала у використанні у NSGA-III дещо іншої реалізації кросовера. Хоча MOEA Framework [56] не є «офіційною» імплементацією, оскільки вихідний код NSGA-III не є публічним, це мабуть найкраще доступне відкрите впровадження коду (розробники співпрацювали з К.Дебом та одним з його учнів для перевірки реалізації).

1.9 Оптимізація «чорного ящика» як динамічна система. Теорія генетичних алгоритмів

У нашому підході для побудови математичної моделі динамічної системи, який спирається на вимірювані дані в досліджуваній системі, зазвичай використовують статистичні методи. Ідентифікація досліджуваної системи включає оптимальний дизайн експериментів для ефективної генерації даних для підгонки моделей, а також редукції самої моделі.

Математичне моделювання використовують у багатьох дисциплінах, наприклад, в інженерних моделях для аналізування існуючого процесу або створення нових процесів. Моделі можна класифікувати кількома способами, але основна класифікація полягає у визначенні трьох різних підходів до математичної моделі, залежно від ступеня апріорного знання про систему, яку ми досліджуємо: модель «білого ящика», модель «чорного ящика» та модель «сірого ящика» [71].

В моделі «білого ящика» доступна вся необхідна інформація. Тут моделі будуються на базі перших принципів (електричні, механічні, хімічні принципи тощо). У цьому виді моделей всі рівняння та параметри можуть бути визначені шляхом теоретичного моделювання.

У моделі «чорного ящика», строго кажучи, немає апіорної інформації про систему, але іноді ми можемо мати деяку попередню інформацію про неї. Тут немає прямого зв'язку між параметрами моделі та першими принципами функціонування системи. По суті, ця модель базується лише на даних вимірювання. Мета моделі «чорного ящика» для формулювання динамічної системи – це описати будь-яку динаміку в системі [87].

1.9.1 Марківські ланцюжки та генетичні алгоритми

Для аналізу генетичних алгоритмів часто використовують марковські ланцюжки як формальну структуру для моделювання генетичного алгоритму [47]. Марковські ланцюжки можна використовувати аби показати, що канонічний генетичний алгоритм без елітності в загальному випадку може не збігатися з оптимальним розв'язком [100]. Тобто, якщо ГА постійно не відбирає найкращого розв'язку в кожному поколінні, існує ймовірність (через кросовер та мутацію), що хромосома, яка відповідає за найкращий розв'язок, буде втрачена. Зауважимо, що ГА без елітарності відповідає алгоритму, що його запропонував Джон Генрі Голланд у 1975 році [59].

Марковські ланцюжки визначаються за допомогою стохастичних процесів. Стохастичний процес – це сукупність випадкових змінних X_i , індексованих деяким набором з множини A , тобто $(X_i: i \in A)$. Якщо A – дискретний набір, то ми маємо дискретний стохастичний процес. Якщо A – неперервна множина, то ми маємо неперервний стохастичний процес. Нехай X_0, X_1, \dots – стохастичний процес. Марковський ланцюжок визначаємо як

послідовність випадкових величин таких, що наступний стан $X_{(i+1)}$ залежить лише від поточного стану X_i (властивості без пам'яті), тобто

$$P(X_{(i+1)} = y | X_i = x_i, \dots, X_0 = x_0) = P(X_{(i+1)} = y | X_i = x_i).$$

Імовірність $P_{xy} = P(X_{(i+1)} = y | X_i = x)$ називається ядром переходу [112]. Зауважимо, що $\sum_y P_{xy} = 1$. Початковий розподіл для X_0 визначає розподіл для будь-якого n -того стану [70].

Генетичні алгоритми можна описати як ланцюжок Маркова, де стан генетичних алгоритмів залежить лише від генів окремих осіб. Окрім того, що марковські ланцюжки були використані для забезпечення формальної структури для аналізу генетичних алгоритмів, вони пов'язані також з масштабованими генетичними алгоритмами, коли всі три оператори збігаються окремо, причому частота мутацій сходиться до строго позитивного значення. Вони описують, що генетичний алгоритм є строго ергодичним. Тому збіжність для генетичного алгоритму – це є збіжність відповідного ланцюжка Маркова до розподілу ймовірності стаціонарного стану, який є ненульовим лише для популяцій, що містять оптимальні індивіди [102].

1.9.2 Проста модель генетичного алгоритму

Теорія генетичних алгоритмів була предметом інтенсивних досліджень протягом останніх десятиліть. Базовою моделлю для генетичного алгоритму, яка дає точне математичне формулювання, є так звана «Проста модель генетичного алгоритму» (ПГА) [114]. Ця модель описує генетичний алгоритм як динамічну систему і добре вивчена в літературі.

У цій моделі генетичний алгоритм представлено як марковський ланцюжок, де стани еволюційної системи є популяціями, а переходами

керують генетичні оператори: селекція, мутація та схрещування (кросовер або кросоверинг) [97]. Мутація гарантує, що марковський ланцюжок є зв'язаним, тому існує унікальний рівноважний розподіл за популяціями. Ймовірність виникнення конкретної популяції в одному поколінні залежить лише від факторів попереднього покоління та зовнішніх впливів. Цей випадковий процес описується ланцюжком Маркова, що характеризується матрицею переходу $T_{\vec{q}, \vec{p}}$ від популяції \vec{p} до популяції \vec{q} , де у векторі популяцій

$$\{\vec{p} = (p_1, p_2, \dots, p_m)^t, \quad 0 \leq p_\alpha \leq 1, \sum_{\alpha=1}^m p_\alpha = 1\},$$

компонента p_α є ймовірністю появи в популяції α -тої особи. Ми розглядаємо популяції з m різними типами індивідів у просторі елементарних подій Ω [89, 92].

Давайте коротко згадаємо результати що отримані в рамках простої моделі генетичного алгоритму, які представлені в [114], і встановимо можливі зв'язки з задачею оптимізації наших параметрів.

Нехай ми маємо популяцію з N різних типів індивідів у просторі елементарних подій Ω . Кожний елемент з Ω можна розглядати як «єдиний індивід» з визначеним значенням функції вартості.

Популяція складається з m -підмножин ($m \ll N$) кожна з яких містить b_{α_i} індивідів типу α_i де $i = 1, \dots, m$ і ми визначили вектор:

$$\vec{b} = (b_{\alpha_1}, b_{\alpha_2}, \dots, b_{\alpha_m})^t,$$

де $\alpha_i \in \Omega$. Розмір популяції є $\bar{m} = \sum_{i=1}^m b_{\alpha_i}$.

Ми можемо визначити вектор популяції в ймовірнісному представленні наступним чином $\vec{p} = (p_1, p_2, \dots, p_N)^t$ де p_α ($p_{\alpha_i} = b_{\alpha_i} / \bar{m}$) є ймовірність

появи α -того індивіда в векторі популяції \vec{b} .

Як ми згадували перед цим раніше повторне застосування генетичного алгоритму дає послідовність векторів $\vec{p} \in \Lambda$ де

$$\Lambda = \{(p_1, p_2, \dots, p_N)^t \in R^N \mid 0 \leq p_\alpha \leq 1, \sum_{\alpha=1}^N p_\alpha = 1\}.$$

Λ є множиною допустимих станів для популяцій яку можна розглядати як $(m - 1)$ -мірний симплекс (гіпер-тетраедр).

Генетичні оператори (селекція, мутація та кросовер) діють у векторному просторі $\Lambda = \{(p_1, p_2, \dots, p_m)^t\}$ векторів популяцій. Генетичний оператор $G_\alpha(\vec{p})$ задає певну ймовірність отримання індивіда α в наступному поколінні, якщо попереднє покоління було зафіксоване вектором \vec{p} , і повний генетичний оператор визначено відображенням $G: \Lambda \rightarrow \Lambda$, де $G(\vec{p}) = \prod_{\alpha \in \Omega} G_\alpha(\vec{p})$, і його можна розглядати як евристичну функцію. $G(\vec{p})$ є насправді композицією трьох різних відображень: селекції, мутації та кросовера.

Визначимо оператор генетичного відбору (селекції) $F: \Lambda \rightarrow \Lambda$, де $F(\vec{p}) = \prod_{\alpha \in \Omega} F_\alpha(\vec{p})$ і α -та компонента $F_\alpha(\vec{p})$ є ймовірністю появи індивіда типу α якщо селекція застосовується до $\vec{p} \in \Lambda$. Оператор селекції вибирає індивідів із поточного покоління за допомогою вектор-функції витрат, $\vec{f} = \{f_\alpha\} \in R^m$, де $f_\alpha = f(\alpha)$, $\alpha \in \Omega$. Цей тип селекції збирає елементи з ймовірністю, пропорційною їх фітнес-функції. Це відповідає евристичній функції:

$$F(\vec{p}) = \frac{\text{diag}(f) \cdot \vec{p}}{\vec{f}^t \cdot \vec{p}},$$

де $\text{diag}(\vec{f})$ матриця з елементами з \vec{f} по діагоналі і з нулями в

інших місцях.

Оператор мутації $U: \Lambda \rightarrow \Lambda$ є матрицею розміру $m \times m$ з дійсними матричними елементами $u_{\alpha, \beta} > 0$ для всіх α, β і $u_{\alpha, \beta}$ представляє ймовірність того, що індивід $\beta \in \Omega$ мутує в індивід $\alpha \in \Omega$. Тоді $(U \cdot \vec{p})_{\alpha}$ є ймовірність появи індивіда типу α після застосування мутації для популяції \vec{p} .

Оператор кросовера (схрещування) визначається як $C: \Lambda \rightarrow \Lambda$, де

$$C(\vec{p}) = (\vec{p}^t \cdot \hat{C}_1 \cdot \vec{p}, \dots, \vec{p}^t \cdot \hat{C}_m \cdot \vec{p})$$

і $\hat{C}_1, \dots, \hat{C}_m$ є послідовність симетричних матриць розміру $N \times N$ з невід'ємними дійсними матричними елементами. Тут $C_{\alpha}(\vec{p}) = \vec{p}^t \cdot \hat{C}_{\alpha} \cdot \vec{p}$ представляє ймовірність того, що індивід α генерується шляхом застосування схрещування до популяції \vec{p} .

Поєднуючи відображення селекції, мутації та кросовера, отримаємо повний оператор G для генетичного алгоритму

$$G: \Lambda \rightarrow \Lambda, \quad G(\vec{p}) = (C \circ U \circ F)(\vec{p}). \quad (1.8)$$

Якщо ми знаємо евристичні функції $G_{\alpha}(\vec{p})$, ми можемо написати матрицю переходу, яка є стохастичною і визначає ймовірність переходу популяції \vec{p} в популяцію \vec{q}

$$T_{\vec{q}, \vec{p}} = \bar{m}! \prod_{\alpha \in \Omega} \frac{(G_{\alpha}(\vec{p}))^{\bar{m}q_{\alpha}}}{(\bar{m}q_{\alpha})!} \quad (1.9)$$

де $G_{\alpha}(\vec{p})$ є ймовірність утворення індивіда α в наступному поколінні і $\bar{m}q_{\alpha}$ є кількість копій індивіда α в популяції \vec{q} та \bar{m} є число індивідів в

популяції [89].

Загальні властивості збіжності еволюції результатів застосування простого генетичного алгоритму до оптимального розв'язку були досліджені у [100]. У праці [102] було показано, що старші власні значення матриці переходу визначають властивості збіжності простого генетичного алгоритму, а саме швидкість збіжності визначається другим за величиною власним значенням після найстаршого власного значення.

Доведення було зроблене для діагоналізованих матриць переходу і узагальнене на матриці, які можна представити в Жордановій нормальній формі.

Для оптимізації симуляції за допомогою GeantV ми ідентифікуємо набір параметрів оптимізації, який є важливим для визначення продуктивності моделювання транспорту частинок (наприклад, розмір вектора транспортованих частинок або інші важливі конструктивні особливості), та побудуємо матрицю даних $X_{\alpha,i} = \{(\vec{x}_{\alpha})_i\} = \{\vec{x}_{\alpha}\}$, яка містить значення цих параметрів. Тут індекс i нумерує параметри оптимізації ($i = 1, \dots, n$), а індекс α нумерує кількість вимірів цих параметрів ($\alpha = 1, \dots, M$ для M вимірів). В рамках генетичного алгоритму індекс α нумерує M індивідуумів, а вектор популяції в представленні даних має вигляд $(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_M)$.

Нагадаємо зв'язок між двома представленнями вектора популяції – представленням через дані та ймовірнісним представленням. В першому випадку ми розглядаємо вектор який отримано з вимірів i -того параметра $\vec{x}_i' = \{(\vec{x}_i')_{\alpha}\} = \{(\vec{x}_i')_1, (\vec{x}_i')_2, \dots, (\vec{x}_i')_M\}$, де компонента $(\vec{x}_i')_{\alpha}$ відповідає значенню i -того параметру в α -тому вимірюванні в векторі популяції $(\vec{x}_1', \vec{x}_2', \dots, \vec{x}_n')$.

У другому випадку нехай $P_i(x)$ є нормована функція розподілу ймовірності вимірів i -того параметру така що

$$\int_{-\infty}^{\infty} dx P_i(x) = 1.$$

Використовуючи попередню стратегію ми пов'язуємо вектор популяції:

$$(\vec{p}_1, \vec{p}_2, \dots, \vec{p}_n) \quad \text{з} \quad (\vec{x}_1', \vec{x}_2', \dots, \vec{x}_n'),$$

де

$$\vec{p}_i = \{(\vec{p}_i)_1, (\vec{p}_i)_2, \dots, (\vec{p}_i)_M\},$$

i компоненти $(\vec{p}_i)_\alpha$ є ймовірністю отримати при вимірюванні i -того параметру значення $(\vec{x}_i')_\alpha$ в α -тому вимірюванні.

Відповідно, по аналогії з (1.8), можемо записати представлення повного генетичного оператора \hat{G} , що діє в векторному просторі даних $X = \{\vec{x}\}$, як поєднання дії представлення операторів селекції \hat{F} , мутації \hat{U} та кросоверу \hat{C} в цьому ж просторі

$$\hat{G} : X \rightarrow X, \quad \hat{G}(\vec{x}) = (\hat{C} \circ \hat{U} \circ \hat{F})(\vec{x}). \quad (1.10)$$

Або для дії повного генетичного оператора \hat{G} на матрицю даних $X_{\alpha,i} = \{(\vec{x}_\alpha)_i\} = \{\vec{x}_\alpha\}$ маємо

$$\hat{G} : \hat{X} \rightarrow \hat{X}', \quad \hat{X}' = \hat{G} \cdot \hat{X} = (\hat{C} \circ \hat{U} \circ \hat{F}) \cdot \hat{X}. \quad (1.11)$$

Розглядаючи генетичні алгоритми як ланцюжок Маркова, можна математично акуратно описати генетичний алгоритм як динамічну систему, що описує еволюцію індивідів у вихідному просторі можливих популяцій фіксованого розміру m , де m – це кількість вимірювань під час експерименту. Для таких динамічних систем можна знайти багато цікавих математичних об'єктів, наприклад, таких як фіксовані точки. Ці об'єкти можуть бути не тільки загальними для звичайних генетичних алгоритмів, але також загальні для задач багатокритеріальної оптимізації.

1.10 Особливі точки та їх роль в генетичному алгоритмі

Фіксовані точки є власними векторами матриці $\hat{U} \cdot \text{diag}(f)$ [52]. Відповідні власні значення – це середні значення функції пристосованості у фіксованій точці. Теорема Перрона-Фробеніуса стверджує, що саме одна з цих фіксованих точок розташована в Λ і це відповідає найбільшому власному значенню. Інші точки розташовано за межами Λ , але вони можуть вплинути на динаміку всередині множини Λ [99].

Аналізуючи фіксовані точки \hat{G} , можна ідентифікувати стани, на які алгоритм витрачає багато часу. Треба враховувати фіксовані точки всередині та за межами Λ , оскільки всі вони можуть мати вплив. Це дасть змогу нам встановити метастабільні стани [98].

Теорема Гейрингера [46, 85] є класичним результатом з генетики популяцій. Вона описує фіксовані точки \hat{C} , задані шаблонами. Набір фіксованих точок – це популяції, що перебувають у рівновазі. Це означає, що якщо вибрати рядок випадковим чином з такої популяції, ймовірність знаходження 1 в будь-якому положенні, не залежить від змісту решти рядка [96]. Теорема Гейрингера також стверджує, що якщо ви починаєте з деякого вектора популяції p і неодноразово застосовуєте \hat{C} , то границею буде одна з

цих фіксованих точок. Ймовірність знаходження 1 в заданій бітовій позиції в цій фіксованій точці буде просто еквівалентною у співвідношенні до 1 у такому положенні в початковій популяції [98].

1.10.1 Поведінка вихідної популяції для скінченних популяцій

Теорема Гейрингера спочатку була доведена на моделі нескінченної популяції. Нещодавно доведено варіант теореми для скінченних (фінітних) популяцій [84].

У нашому варіанті ми розглядаємо стаціонарний розподіл ланцюжка Маркова. Вибрана випадковим чином популяція з цього розподілу, а також випадковим чином вибраний рядок, показує що ймовірність знаходження одиниці в будь-якому заданому положенні залежить лише від частки одиниць у такому положенні в первинній популяції.

Припустимо, що у нас є генетичний алгоритм з пропорційним відбором та кросовером. Тоді $\hat{G} = \hat{C} \cdot \hat{F}$. Однорідні популяції є фіксованими точками, а також поглинальними станами марковського ланцюжка [50]. Однак, можуть існувати і інші фіксовані точки. Припускається, що єдиними асимптотично стійкими фіксованими точками є однорідні популяції. Доведення цього припущення становить нерозв'язану задачу [98].

1.10.2 Фіксовані точки та локальні оптимуми

У загальному випадку можна розділити фіксовані точки на стійкі та нестійкі з відповідним ступенем нестійкості, причому ступінь нестійкості визначається відповідною поведінкою збурення в околі фіксованої точки при дії на збурення рівнянь динамічної системи. З урахуванням цього маємо, що деякі однорідні популяції асимптотично стабільні, а деякі є нестійкими.

Припустимо, що у нас є асимптотично стійка популяція, яка містить лише копії бітового рядка \mathbf{k} . Тоді, якщо врахувати пристосованість \mathbf{k} і пристосованість всіх його сусідів, то ми знаходимо, що асимптотично стабільні однорідні популяції містять локальні оптимуми.

Найкраща вихідна популяція в ГА визначається тим, що стохастичні відображення мають властивості збіжності, які відповідають дискретним незвідним марковським процесам: збіжності до унікального стаціонарного розподілу ймовірностей. Оскільки такі розподіли відповідають фіксованим точкам моделі нескінченної популяції, то фіксовані точки є фундаментальними об'єктами, цікавими для дослідження. Поведінка відображення \mathbf{G} в околі фіксованої точки \mathbf{x}^* визначається власними значеннями лінеаризованого відображення $\mathbf{LG}(\mathbf{x})$.

Розглянемо ГА як дискретну динамічну систему що описує еволюцію вектору популяції в ймовірносному представленні:

$$\vec{p}_{k+1} = \mathbf{G}[\vec{p}_k]$$

або

$$\vec{p}_{k+1} = \mathbf{G}^k[\vec{p}_0], \quad \mathbf{G}^k = \mathbf{G} \circ \mathbf{G} \circ \dots \circ \mathbf{G},$$

Фіксована точка в цьому випадку визначається наступним чином:

$$\vec{p}^* = \mathbf{G}[\vec{p}^*].$$

Фіксовані точки для ГА були досліджені в монографіях Рівза та Роу (2003) [98]. Гедеон (2010) [58] показав, що ГА має кінцевий набір гіперболічних фіксованих точок.

Тоді поведінка відображення \mathbf{G} в околі фіксованої точки $[\vec{p}^*]$ визначається власним значенням лінеаризованого відображення $\mathbf{LG}[\vec{p}^*]$

$$\Delta p'_i = p'_i - p_i^* = \frac{\partial G_i}{\partial p_j} [\vec{p}^*] (p_j - p_j^*) = LG_{i,j} [\vec{p}^*] \Delta p_j.$$

Якщо всі власні значення LG мають абсолютне значення менше одиниці, то всі ітерації, що починаються в околі \vec{p}^* будуть збігатися до \vec{p}^* . Якщо існує принаймні одне власне значення з абсолютним значенням більше одиниці, то майже всі ітерації будуть відхилятися від \vec{p}^* . Для гіперболічних фіксованих точок LG не має власних значень, що лежать на одиничному колі в комплексній площині.

Висновки до розділу 1

1. Проведено аналіз програмового забезпечення, яке використовується для симуляції (моделювання) проходження елементарних частинок через матерію при оброблюванні експериментів в ЦЕРН на ВАК, зокрема, системи автоматизованого проектування і розрахунку нового покоління GeantV. Аналіз дав змогу визначити, як головні чинники, такі як місткість пам'яті, обсяг даних, час роботи, кількість інструкцій для функції пристосованості, кількість фізичних подій та кількість фізичних подій в буфері, кількість рядків та кількість препаративів впливають на продуктивність обчислень та можуть бути використані для стохастичної оптимізації ефективності роботи системи автоматизованого проектування і розрахунку симуляції в GeantV, а також виявити слабкі місця у функціональній моделі GeantV.

2. Сформульовано багатокритеріальну задачу оптимізації продуктивності системи автоматизованого проектування і розрахунку симуляції транспорту частинок GeantV, обґрунтовано вибір генетичного алгоритму в якості базової процедури такої оптимізації і визначено простір розв'язків та цільовий простір. Використовуючи цей аналіз, зокрема,

представлення генетичного алгоритму як динамічної системи, використовуючи його зв'язок з марковськими ланцюжками та аналіз властивостей власних чисел матриці переходу, вдалося визначити для цієї багатокритеріальної задачі оптимізації роль Парето-фронту, оптимальної Парето точки та нерухомої точки [89].

3. Визначено, що не існує єдиного рецепту для вибору стратегії правильного алгоритму для рішення багатокритеріальної задачі оптимізації продуктивності системи автоматизованого проектування і розрахунку GeantV, але генетичні алгоритми є найбільш придатними для цієї задачі та широко використовуються для такого типу завдань як багатокритеріальні задачі оптимізації.

РОЗДІЛ 2

МЕТОД НЕЗЦЕНТРОВАНИХ ГОЛОВНИХ КОМПОНЕНТ ЯК ОСНОВА ДЛЯ НОВОГО ОПЕРАТОРА В ГЕНЕТИЧНИХ АЛГОРИТМАХ

Високопродуктивні обчислення у ФВЕ використовують велику кількість паралельних додатків із сотнями тисяч симуляцій транспорту елементарних частинок. При великих об'ємах обчислень, гостро постає питання вибору алгоритму для оптимізації роботи паралельних додатків. Зазвичай для задач багатокритеріальної оптимізації використовуються генетичні алгоритми або еволюційні стратегії. Але необхідно зауважити, що різниця між застосуванням генетичних алгоритмів та еволюційних стратегій для моделювання сучасних високоємних задач транспорту частинок в сучасних детекторах експериментів на ВАК може бути значною, і головне питання полягає в тому, як швидко і ефективно ми можемо знайти оптимальні розв'язки для налаштування великого набору програм симуляцій транспорту частинок.

Виходячи з наявних підходів до оптимізації програмових ресурсів у складних симуляціях у ФВЕ, відмітимо статтю [74], в якій розглянуто підхід, заснований на застосуванні генетичних алгоритмів. Але цей підхід дуже важко адаптувати для використання у нашій конкретній задачі оптимізації роботи додатків САЕ-системи GeantV [23]. У нашому випадку оптимізації, додатки GeantV мають значний набір параметрів, для яких нам потрібно знайти оптимальні значення, використовуючи методи багатокритеріальної оптимізації для проблеми «чорного ящика».

Генетичні алгоритми і стратегії еволюції належать до одного і того самого класу еволюційних методів, які базуються на підходах стохастичного пошуку або методах оптимізації «чорного ящика». Різниця полягає лише в послідовності застосовування операторів та важливості використання тих чи

інших операторів для збільшення ймовірності відтворення індивідів у поколіннях – для генетичного алгоритму використовують оператори селекції, кросоверу та мутації, тоді як в еволюційних стратегіях головним оператором здебільшого є мутація (через мутаційну матрицю коваріантності).

Однією із задач представлення генетичного алгоритму через марковські ланцюжки є визначення еволюції у відповідному напрямі для швидкого наближення до оптимального розв'язку. Збіжність роботи генетичних алгоритмів була доведена за допомогою зв'язку генетичних алгоритмів з марковськими ланцюжками. У праці [111], використовуючи цей зв'язок, доведено, що при застосуванні генетичного алгоритму популяція асимптотично приходить до рівномірної популяції, що і дає оптимальний розв'язок для задачі.

Ідея нашого методу полягає у застосуванні змішаного підходу, який може привести до керованої збіжності генетичного алгоритму до справжнього Парето-фронту.

2.1 Редукція цільових функцій в задачах багатокритеріальної оптимізації

Використання багатокритеріальних алгоритмів у задачах оптимізації, що мають набір складних цільових функцій, супроводжується складністю як обчислювального, так і методичного характеру, а саме:

- великою обчислювальною складністю (при збільшенні кількості цільових функцій розмірність фронту Парето);
- низькою масштабованістю існуючих алгоритмів багатокритеріальної оптимізації;
- проблемою забезпечення візуалізації оптимального фронту Парето.
- одним з підходів для подолання цих труднощів є зменшення

(редукція) кількості цільових функцій, що може збільшити малу швидкість збіжності алгоритмів. Коротко нагадаємо сучасні підходи у вивченні редукції цільових функцій як в лінійних, так і нелінійних випадках. Відмітимо наступні: а) збереження відношення домінування (ЗВД); б) вибір неконтрольованих особливостей; в) вибір кута Парето; та г) редукції цільових функцій на основі машинного навчання.

2.1.1 Метод «збереження відношення домінування»

Головна ідея підходу «збереження відношення домінування» (ЗВД), запропонованого Е. Зицлером [26], полягає або в знаходженні мінімуму множини цільових функцій з відповідною похибкою (ϵ -MOSS задача – англ. minimum objective subset problem), як мірою зміни у відношенні домінування, або в знаходженні множини цільових функцій, яка має спеціальний розмір з попередньо встановленою помилкою (k-EMOSS minimum objective subset problem задача). Обидва підходи розглядають як NP-складні.

ЗВД-підхід фактично ґрунтується на збереженні відношення домінування в заданих недомінантних розв'язках при редукції і впливає з наступного критерію. Якщо із заданої множини цільових функцій даної задачі одну з цільових функцій видаляють, а відношення домінування між цільовими векторами залишаються незмінними, то вважається, що вона не конфліктує з іншими цільовими функціями.

Підхід «вибору неконтрольованих особливостей» описує кореляції між цільовими функціями та, використовуючи їхні значення як ефективну відстань в просторі цільових функцій, виділяє конфліктні цільові функції, а також множину цільових функцій розділяє на групи спеціального розміру, вибираючи їх таким чином, щоб найкомпактніше розташовані сусіди потрапили до групи. Редукція в цьому підході полягає у виключенні центру

такої групи (найменш скорельованих цільових функцій в групі), і тут, як і в попередньому підході, треба мати критерій точки зупинки дії алгоритму, наприклад для алгоритмів багатокритеріальної оптимізації, за умови, якщо його виберуть в задачах ϵ -MOSS чи k-EMOSS (minimum objective subset problem).

Вперше цей підхід був запропонований в статті [79], в якій автори використали алгоритм [86]. У статті [86] в якості метрики, що вимірює подібність особливостей, було вибрано кореляційну метрику між двома випадковими величинами, що визначає міру корельованості між випадковими особливостями. Такий підхід дозволив вимірювати від'ємні кореляції між цільовими функціями, які визначають конфліктні цільові функції і становлять множину цільових функцій, яка може включати фронт Парето.

Графічно ми можемо розділити множину цільових функцій на майже однорідні групи сусідів розміром q навколо кожної цільової функції. Мірою ефективною відстані в просторі цільових функцій для цього підходу є ступінь конфліктності двох цільових функцій. Чим більше вони конфліктують між собою, то далі вони одна від одної.

У рамках цього підходу ми виберемо групу найкомпактніше розташованих сусідів, потім зафіксуємо її центр та зредукуємо найближчих до центру q -сусідів (це цільові функції, які найменше конфліктують між собою в групі). Тут q означає ефективний розмір околу найближчих сусідів і точку зупинки алгоритму. Для реалізації цього підходу треба знайти матрицю кореляцій між конфліктними цільовими функціями та їхніми найближчими q сусідами.

2.1.2 Еволюційний алгоритм пошуку кута Парето

З огляду на складність обчислювання, працювати з наближенням всього фронту Парето важко і майже неможливо при використанні великого числа

поколінь та великих розмірів популяцій. Всі розв'язки повинні бути близькими до фронту Парето так, щоб вони могли зафіксувати розмірність фронту та бути поширені на весь фронт Парето, щоб забезпечити знаходження всіх необхідних цільових функцій [107].

У алгоритмі «вибір кута Парето» ми припускаємо, що можна знайти достатню кількість розв'язків, за допомогою яких можна визначити розмірність Парето-фронту та його границі.

Підхід «вибір кута Парето» є специфічним алгоритмом редукції [107], навіть при наближеному знаходженні фронту Парето, кількість розв'язків зростає експоненціально із збільшенням кількості цільових функцій. Якщо використовувати NSGA-II як типового представника алгоритмів багатокритеріальної оптимізації, то при розв'язанні цієї задачі нам потрібно буде збільшити кількість поколінь в рамках генетичного алгоритму, що призведе до збільшення обсягу розрахунків фітнес-функцій, а у випадку моделювання транспорту частинок за допомогою GeantV – до витратних обчислювань.

2.1.3 Спосіб зменшення кількості цільових функцій на основі методу головних компонент

У праці Деба і Саксени [38] запропоновано спосіб зменшення кількості цільових функцій на основі методу головних компонент. Головна ідея полягає в тому, що якщо дві цільові функції мають від'ємне значення кореляційної функції, то вони є конфліктними і включаються в множину даних (матрицю даних) для аналізу фронту Парето. Для аналізу цієї множини даних (цільових функцій) та її подальшої редукції автори використали метод головних компонент (власні вектори кореляційної матриці). Вибираючи у старшому власному векторі компоненти з найбільшим від'ємним значенням та з найбільшим додатнім значенням, автори змогли виділити дві найконфліктніші

цільові функції.

Далі автори досліджували власні вектори, які йдуть за старшим власним вектором, причому вибирали такі, щоб їхні власні значення у сумі перевищували порогове значення. Тоді для використання цієї ідеї у будь-якому алгоритмі, наприклад, в NSGA-II процедури розглядають як ітеративний процес, і отриману множину цільових функцій редукують за допомогою кореляційного аналізу. Фактором зупинки ітераційного процесу є умова, коли поточна підмножина дорівнює підмножині, сформованій в попередній ітерації. Недоліком цього методу є те, що ми не можемо гарантувати збереження структури домінування.

В наступному параграфі ми більш детально розглянемо метод головних компонент, який в подальшому є основою для визначення нового оператора редукції цільових параметрів в генетичному алгоритмі.

2.2 Метод головних компонент

Метод головних компонент [69, 117] – це метод статистичного аналізу спостережуваних даних, в якому ортогональні перетворення використовують для трансформації набору даних зкорельованих змінних у набір значень лінійно не зкорельованих змінних, які називаються головними компонентами. Це перетворення визначається таким чином, що перша головна компонента має найбільшу дисперсію, і кожна наступна компонента, у свою чергу, має найбільшу дисперсію за умови, що вона ортогональна до попередніх компонентів. Отримані вектори, що відповідають головним компонентам, утворюють незкорельований ортогональний базис. Цей метод контрольованим чином уможливорює редукцію змінних в статистичному аналізі спостережуваних даних, нехтуючи головними компонентами з малими значеннями дисперсії.

Для ефективного застосування методу головних компонент зазвичай використовують метод сингулярного представлення матриці даних, який дає змогу значно спростити обчислення, а також узагальнення методу головних компонент, яке для нашого дослідження є дуже важливим з огляду на застосування для незцентрованої матриці даних.

Загально прийнято, що найраніші варіанти методу, відомого тепер як метод головних компонент, сформулювали у своїх працях Пірсон (Pearson, 1901) [94] та Хотеллінг (Hotelling, 1933) [63]. Пірсон досліджував задачу знаходження ліній та площин, найбільш наближених до деякої множини точок в p -вимірному просторі (так звана задача геометричної оптимізації), за допомогою методу що нагадує метод головних компонент.

Хотеллінг досліджував, чи можна знайти менший «фундаментальний набір незалежних змінних, які визначають значення» максимального вкладу в статистичні величини вихідних p змінних. У своїй праці він зазначає, що в літературі з психології такі змінні називаються «факторами», і щоб уникнути плутанини, пов'язаної з використання слова «фактор» у математиці, ввів альтернативний термін «компоненти».

Хотеллінг вибирає ці компоненти так, щоб максимізувати їхні послідовні внески в повну дисперсію вихідних змінних, і називає такі компоненти «головними компонентами». Алгебраїчний аналіз, який оперує з такими компонентами, був сформульований в термінах, близьких до сучасних, і отримав назву «метод головних компонент».

Для знаходження головних компонент Хотеллінг використовував підхід, що був подібний до методу Лагранжових множників, і його задача зводилася до задачі знаходження власних векторів та власних значень, але його підхід мав три відмінності. По-перше, він оперував з кореляційною матрицею, а не з матрицею коваріацій, як в сучасних підходах. По-друге, він розглядав вихідні змінні як лінійні функції компонент, і не використовував визначення компонент як лінійних функцій вихідних змінних. По-третє, він не

використовував матричну техніку.

Тепер метод головних компонент (МГК, Principle Component Analysis, PCA) широко використовують в різних галузях, де застосовують методи аналізу даних. Виокремимо три ключові сучасні модифікації цього методу: Sparse Principle Component Analysis [32], Robust Principle Component Analysis [119] та Kernel Principle Component Analysis [103].

Зазвичай метод головних компонент використовують для аналізу матриці коваріації, щоб зменшити розмірність складного набору даних (у нашому випадку, це набір даних з вимірювання продуктивності роботи системи автоматизованого проектування і розрахунку GeantV). МГК традиційно застосовується до центрованої матриці даних. Ми покажемо, як можна застосувати МГК до незцентрованої матриці даних, і що цей підхід є дуже зручним для зменшення розмірності складного набору даних при його застосуванні до задач оптимізації, які розв'язуються за допомогою генетичних алгоритмів.

Перші кроки для знаходження зв'язку між МГК для центрованої та незцентрованої матриць даних були зроблені в [68, 60].

У цьому підрозділі ми розвиваємо математичний апарат застосування МГК для незцентрованої матриці даних з метою його використання для модифікації генетичних алгоритмів, що може прискорити розв'язання багатокритеріальних задач оптимізації за допомогою цих методів [92]. Для задачі оптимізації системи автоматизованого проектування і розрахунку GeantV ми розглядаємо матриці даних, що складаються із значень вимірюваних параметрів продуктивності системи автоматизованого проектування і розрахунку, тоді як цільові функції є набором функцій придатності, що оцінюють ефективність роботи симуляцій системи автоматизованого проектування і розрахунку.

2.3 Метод головних компонент для центрованої матриці даних

Для оптимізації роботи системи автоматизованого проектування і розрахунку GeantV при обробці експериментальних даних ми визначаємо параметри оптимізації, які є важливими для проведення симуляцій транспорту елементарних частинок (наприклад, розмірність вектору, що зберігає інформацію про характеристики частинок, та інші), і визначаємо матрицю даних \hat{X} розміром $M \times n$

$$\hat{X} = \{X_{\alpha,i}\} = \{(\vec{x}_{\alpha})_i\} = \{\vec{x}_{\alpha}\}, \quad (2.1)$$

яка є незцентрованою матрицею даних і містить значення вимірювань цих параметрів. У цій матриці індекс i нумерує параметри оптимізації ($i = 1, \dots, n$), а індекс α нумерує кількість вимірювань параметрів ($\alpha = 1, \dots, M$ для M вимірювань). В рамках формулювання генетичного алгоритму матриця даних описується через M -вибірок даних з n -вимірному простору. У цьому випадку M – це кількість індивідів в популяції, n – розмір індивіда (n є розмірність вектору генів $\vec{x} = \{x_i\}$), і вектор популяції визначаємо через $(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_M)$.

Визначимо центровану матрицю даних $\hat{X}^{(c)}$

$$\hat{X}^{(c)} = \{X_{\alpha,i}^{(c)}\} = \{X_{\alpha,i}^{(u)} - \mu_i\}, \quad (2.2)$$

де μ_i – середнє за M -індивідуумів i -тої компоненти індивідуума

$$\mu_i = \frac{1}{M} \sum_{\alpha=1}^M X_{\alpha,i}^{(u)}, \quad 1 \leq i \leq n, \quad \vec{\mu} = \{\mu_i\}. \quad (2.3)$$

Центрована матриця даних визначає матрицю коваріації $\hat{\Sigma}$

$$\hat{\Sigma} = \frac{1}{M} \hat{X}^{(c) t} \cdot \hat{X}^{(c)} = \{\Sigma_{i,j}\} = \frac{1}{M} X_{i,\alpha}^{(c)t} X_{\alpha,j}^{(c)}, \quad (2.4)$$

де в добутку матриць за індексами, що повторюються, мається на увазі сумування.

Мотивацію застосування методу головних компонент можна сформулювати у вигляді наступної задачі. Нехай у нас є вхідна матриця даних $X_{\alpha,j}^{(c)}$ рангу n (прямокутна матриця розміру $M \times n$), тоді можна поставити питання – як зробити контрольоване наближення (з явно обчислюваною похибкою) цієї матриці за допомогою вихідної матриці даних $\tilde{X}_{\alpha,j}^{(c)}$ рангу $m < n$ (прямокутна матриця розміру $M \times n$ але з «редукованою розмірністю даних»).

Зрозуміло, що будь-яке наближення може працювати, коли є малий параметр, що контролює наближення. З формулювання задачі випливає, що такий параметр може існувати при розглядуванні задачі в деякому допоміжному просторі.

2.3.1 Підхід, що базується на максимізації дисперсії

У стандартному методі головних компонент основними об'єктами є ортонормований вектор \vec{u} ($u_i, i = 1, \dots, n$), проєкції $x_{u,\alpha}^{(c)}$ векторів $\vec{x}_\alpha^{(c)} = \vec{x}_\alpha - \vec{\mu}$, $\alpha = 1, 2, \dots, M$ на вектор \vec{u} , які і є компонентами вектора центрованої головної компоненти $\vec{v}^{(c)}$:

$$(\vec{v}^{(c)})_\alpha = x_{u,\alpha}^{(c)} = \vec{x}_\alpha^{(c) t} \cdot \vec{u} = \sum_{i=1}^n X_{\alpha,i}^{(c)} u_i, \quad \vec{u}^t \cdot \vec{u} = 1,$$

та дисперсія головної компоненти σ_v^2

$$\sigma_v^2 = \frac{1}{M} \sum_{\alpha=1}^M \left[\sum_{i=1}^n X_{\alpha,i}^{(c)} u_i \right]^2.$$

Першу головну компоненту $\vec{v}_1^{(c)}$, визначаємо як лінійну комбінацію, що має максимальну дисперсію $v_{\alpha,1}^{(c)} = \vec{x}_\alpha^{(c)T} \cdot \vec{u}_1 = X_{\alpha,i}^{(c)} (\vec{u}_1)_i$, де n -вимірний вектор $\vec{u}_1 = \{(\vec{u}_1)_1, \dots, (\vec{u}_1)_n\}^T$ є розв'язком задачі на екстремум

$$\vec{u}_1 = \arg \max_u \text{Var}(\vec{x}_\alpha^{(c)T} \cdot \vec{u}_1), \quad \vec{u}_1^T \cdot \vec{u}_1 = 1.$$

Друга головна компонента є лінійною комбінацією, що має другу за величиною найбільшу дисперсію, і є ортогональною до першої головної компоненти і т.д.

Визначимо вектор головних компонент $\vec{v}_i^{(c)} = \{(\vec{v}_i^{(c)})_\alpha\}$ ($i = 1, 2, \dots, n$) як

$$(\vec{v}_i^{(c)})_\alpha = X_{\alpha,j}^{(c)} (\vec{u}_i)_j. \quad (2.5)$$

Аби знайти головні компоненти, зручно розглянути варіаційну задачу на екстремум дисперсії як функціонала вектора \vec{u}

$$\sigma_v^2 = \text{Var}(\vec{v}^{(c)}) = \frac{1}{M} \vec{u}^T \cdot \hat{X}^{(c)T} \cdot \hat{X}^{(c)} \cdot \vec{u} = \vec{u}^T \cdot \hat{\Sigma} \cdot \vec{u}. \quad (2.6)$$

Варіаційна задача має Лагранжіан

$$L = \vec{u}^T \cdot \hat{\Sigma} \cdot \vec{u} + \lambda(\vec{u}^T \vec{u} - 1).$$

Умова стаціонарності дає

$$\frac{\partial L}{\partial \vec{u}} = 2\hat{\Sigma} \cdot \vec{u} - 2\lambda \vec{u} = \mathbf{0}, \quad \hat{\Sigma} \cdot \vec{u} = \lambda \vec{u}.$$

Нехай ми маємо n розв'язків останнього матричного рівняння

$$\hat{\Sigma} \cdot \vec{u}_j = \lambda_j^{(c)} \vec{u}_j, \quad 1 \leq j \leq n,$$

тоді \vec{u}_j є власними векторами матриці коваріації $\hat{\Sigma}$ з власними значеннями $\lambda_j^{(c)}$ і \vec{u}_j і задовольняють умову ортогональності

$$\vec{u}_i^t \cdot \vec{u}_j = \delta_{ij}, \quad 1 \leq i, j \leq n, \quad (2.7)$$

та

$$\vec{u}_j^t \cdot \hat{\Sigma} \cdot \vec{u}_j = \lambda_j^{(c)}. \quad (2.8)$$

Тоді з (2.6) ми отримуємо:

- для дисперсії i -тої центрованої головної компоненти

$$\text{Var}(\vec{v}_i^{(c)}) = \vec{u}_i^t \cdot \hat{\Sigma} \cdot \vec{u}_i = \lambda_i^{(c)}. \quad (2.9)$$

Звідси вектор, вздовж якого ми маємо максимальну дисперсію (перша головна компонента) є власним вектором з найбільшим власним значенням матриці коваріації $\hat{\Sigma}$. Власний вектор для другого найбільшого за величиною власного значення визначає другу головну компоненту і т.д.

- для коваріації i -тої та j -тої центрованих головних компонент ($i \neq j$)

маємо

$$\mathbf{Cov}(\vec{v}_i^{(c)}, \vec{v}_j^{(c)}) = \vec{u}_i^t \cdot \hat{\Sigma} \cdot \vec{u}_j = \mathbf{0}. \quad (2.10)$$

Визначимо матрицю $U_{i,j} = \vec{u}_j = (\mathbf{u}_j)_i$, яка з співвідношення (2.7) задовольняє умову ортогональності

$$U_{i,i'}^t U_{i',j} = \delta_{i,j}. \quad (2.11)$$

Звідси в матричній формі одержуємо

$$\hat{U}^t \cdot \hat{\Sigma} \cdot \hat{U} = \hat{\Lambda}^{(c)}, \quad \Lambda_{i,j}^{(c)} = \lambda_i^{(c)} \delta_{i,j}, \quad (2.12)$$

Визначимо матрицю головних компонент $V_{\alpha,j}^{(c)} = (\vec{v}_j^{(c)})_\alpha$

$$V_{\alpha,j}^{(c)} = X_{\alpha,i}^{(c)} U_{i,j}, \quad 1 \leq \alpha \leq M, \quad (2.13)$$

або

$$X_{\alpha,i}^{(c)} = V_{\alpha,j}^{(c)} U_{j,i}^t. \quad (2.14)$$

Для узагальнення припустимо, що матриця коваріації має n власних значень, з них маємо m перших власних значень таких, що задовольняють оцінку

$$\frac{\sum_{i=1}^m \lambda_i^{(c)}}{\sum_{i=1}^n \lambda_i^{(c)}} = \mathbf{0,95}. \quad (2.15)$$

Тоді ми можемо використовувати два підходи для оцінювання наближення:

– наближення, регульоване параметром головних компонент, в якому ми нехтуємо головними компонентами $\vec{v}_{m+1}^{(c)}, \vec{v}_{m+2}^{(c)}, \dots, \vec{v}_n^{(c)}$ в матриці $V_{\alpha,j}^{(c)} = \{\vec{v}_j^{(c)}\} = (v_j^{(c)})(\vec{v}_i^{(c)})_{\alpha}$, де $\vec{v}_j^{(c)}$ є j -та центрована головна компонента. Це можна зробити за умови $\text{Var}(\vec{v}_k^{(c)}) \ll 1$ для $k = m+1, m+2, \dots, n$. Тоді при цьому наближенні можемо покласти $\vec{v}_{m+1}^{(c)} = \vec{v}_{m+2}^{(c)} = \dots = \vec{v}_n^{(c)} = \mathbf{0}$ в $V_{\alpha,j}^{(c)}$ і отримати нову матрицю $\tilde{V}_{\alpha,j}^{(c)}$ розміру $M \times n$, але рангу m ;

– наближення, регульоване параметром власних значень, в якому ми нехтуємо власними значеннями $\lambda_{m+1}^{(c)}, \lambda_{m+2}^{(c)}, \dots, \lambda_n^{(c)}$ в матриці власних значень $\Lambda_{i,j}^{(c)}$. Ми можемо виконати цю процедуру за умови $\lambda_k^{(c)} \ll 1$ для $k = m+1, m+2, \dots, n$. Ця процедура дає нову матрицю власних значень $\tilde{\Lambda}_{i,j}^{(c)}$ розміру $n \times n$ та рангу m з $\lambda_{m+1}^{(c)} = \lambda_{m+2}^{(c)} = \dots = \lambda_n^{(c)} = 0$.

Ми застосуємо наближення, регульоване «параметром головних компонент» до (2.13) і одержимо вихідну матрицю даних $\tilde{X}_{\alpha,j}^{(c)}$ меншого рангу m :

$$\tilde{X}_{\alpha,i}^{(c)} = \tilde{V}_{\alpha,j}^{(c)} U_{j,i}^t = V_{\alpha,1}^{(c)} U_{1,i}^t + \dots + V_{\alpha,m}^{(c)} U_{m,i}^t. \quad (2.16)$$

Для застосування наближення, регульованого «параметром власних значень», зручно використовувати сингулярне представлення (Singular Value Decomposition, SVD) для матриці даних.

2.3.2 Підхід, що базується на сингулярному представленні матриці даних

Коротко нагадаємо сингулярне представлення для матриці \hat{X} (матриця даних розміру $M \times n$ та рангу r , де без втрати загальності покладемо $M \leq$

n , звідси $r \leq n$).

Сингулярне представлення матриці \hat{X} можна записати у вигляді $\hat{X} = \hat{V}\hat{S}\hat{U}^T$, де \hat{V} – матриця розміру $M \times n$, матриця \hat{S} – $n \times n$ діагональна матриця і \hat{U}^T – матриця розміру $n \times n$. Рядки матриці $\hat{V} = V_{\alpha,i}$ позначимо як \vec{v}_i . Вони мають назву ліві сингулярні вектори і утворюють ортонормований базис таким чином, що виконується умова $\vec{v}_i \cdot \vec{v}_j = 1$ для $i = j$, та $\vec{v}_i \cdot \vec{v}_j = 0$ для інших випадків.

Стовпчики матриці \hat{U}^T позначимо як \vec{u}_i , вони є правими сингулярними векторами і також утворюють ортонормований базис. Матричні елементи діагональної матриці \hat{S} називаються сингулярними значеннями $\hat{S} = \text{diag}(s_1, \dots, s_n)$, $s_k > 0$ для $1 \leq k \leq r$, та $s_i = 0$ для $(r + 1) \leq k \leq n$. Для зручності сингулярні вектори упорядковуються за допомогою сортування від більших сингулярних значень до менших сингулярних значень таким чином, щоб найбільше сингулярне значення було у верхньому лівому куті матриці \hat{S} . Зауважимо, що для квадратної симетричної матриці \hat{X} сингулярне представлення є еквівалентним процедурі діагоналізації цієї матриці або розв'язанню задачі знаходження її власних значень.

Для матриці центрованих головних компонент $V_{\alpha,j}^{(c)} = (\vec{v}_j^{(c)})_\alpha$ із співвідношень (2.11) – (2.13) маємо:

$$V_{i,\alpha}^{(c)} {}^t V_{\alpha,j}^{(c)} = M \Lambda_{i,j}^{(c)} = M \lambda_i^{(c)} \delta_{i,j}. \quad (2.17)$$

Визначимо матрицю ортонормованих центрованих головних компонент $\bar{V}_{\alpha,j} = (\bar{v}_j^{(c)})_\alpha$

$$V_{\alpha,j}^{(c)} = \sqrt{M} \bar{V}_{\alpha,i}^{(c)} \Lambda_{i,j}^{(c) 1/2}, \quad (2.18)$$

де

$$\Lambda_{i,j}^{(c) 1/2} = \left[\lambda_i^{(c)} \right]^{\frac{1}{2}} \delta_{i,j},$$

і звідси та з урахуванням (2.14) отримуємо:

$$X_{\alpha,i}^{(c)} = \sqrt{M} \bar{V}_{\alpha,k}^{(c)} \Lambda_{k,j}^{(c) \frac{1}{2}} U_{j,i}^t$$

Тоді із співвідношень (2.17), (2.18) маємо:

$$\bar{V}_{i,\alpha}^{(c) t} \bar{V}_{\alpha,j}^{(c)} = \delta_{i,j},$$

Використовуючи (2.18) та (2.12) – (2.14) легко показати, що виконується співвідношення:

$$\bar{V}_{i,\alpha}^{(c) t} X_{\alpha,k}^{(c)} X_{k,\beta}^{(c) t} \bar{V}_{\beta,j}^{(c)} = \Lambda_{i,j}^{(c)} = \lambda_i^{(c)} \delta_{i,j}.$$

Звідси $\bar{V}_{\alpha,j}^{(c)}$ є матрицею власних векторів $(\bar{v}_j^{(c)})_\alpha$ для матриці Грама:

$$\hat{K}^{(c)} = \hat{X}^{(c)} \cdot X^{(c) t} \text{ розміру } M \times M$$

$$K_{\alpha,\beta}^{(c)} (\bar{v}_j^{(c)})_\beta = X_{\alpha,k}^{(c)} X_{k,\beta}^{(c) t} (\bar{v}_j^{(c)})_\beta = \lambda_j^{(c)} (\bar{v}_j^{(c)})_\alpha$$

і ці вектори співпадають з ортонормованими центрованими головними компонентами.

Нагадаємо, що ортонормовані вектори \vec{u}_j є власними векторами матриці коваріації:

$$\hat{\Sigma} \cdot \vec{u}_j = \Sigma_{i,k} (\vec{u}_j)_k = \frac{1}{M} X_{i,\alpha}^{(c) \ t} X_{\alpha,k}^{(c)} (\vec{u}_j)_k = \lambda_j^{(c)} \vec{u}_j, \quad 1 \leq j \leq n, \quad (2.19)$$

Співвідношення (2.14) визначає сингулярне представлення центрованої матриці даних:

$$X_{\alpha,i}^{(c)} = \sqrt{M} \bar{V}_{\alpha,i}^{(c)} \Lambda_{ij}^{(c) \ 1/2} U_{j,i}^t, \quad (2.20)$$

Таким чином (2.20) зручно використовувати для застосування наближення, регульованого «параметром власних значень». Припустимо, що виконуються умова $\lambda_k^{(c)} \ll 1$ для $k = m + 1, m + 2, \dots, n$, тоді використовуючи це наближення, отримуємо для вихідної матриці даних $\tilde{X}_{\alpha,j}^{(c)}$, ранг якої вже є m

$$\begin{aligned} \tilde{X}_{\alpha,i}^{(c)} &= \sqrt{M} \bar{V}_{\alpha,i}^{(c)} \tilde{\Lambda}_{ij}^{(c) \ 1/2} U_{j,i}^t = \\ &= \sqrt{M} \left(\left[\lambda_1^{(c)} \right]^{\frac{1}{2}} \bar{V}_{\alpha,1}^{(c)} U_{1,i}^t + \dots + \left[\lambda_m^{(c)} \right]^{\frac{1}{2}} \bar{V}_{\alpha,m}^{(c)} U_{m,i}^t \right). \end{aligned} \quad (2.21)$$

Таким чином, ми отримали наближення матриці $X_{\alpha,i}^{(c)}$ за допомогою матриці $\tilde{X}_{\alpha,i}^{(c)}$, що має менший ранг m . Перетворення матриці даних при переході в базис головних компонент називається дискретним перетворенням Кархунена-Льове або перетворенням Хотеллінга [63].

Для оцінювання похибки наближення, регульованого «параметром власних значень», зручно обчислити середньоквадратичну похибку,

використовуючи сингулярне представлення для центрованої матриці даних (2.20)

$$\begin{aligned}
 \eta_m &= \frac{1}{nM} \sum_{\alpha=1}^M \sum_{i=1}^n (X_{\alpha,i}^{(c)} - \tilde{X}_{\alpha,i}^{(c)})^2 = \\
 &= \frac{1}{nM} \sum_{\alpha=1}^M \sum_{i=1}^n \left(\sqrt{M} \sum_{k=m+1}^n \sqrt{\lambda_k^{(c)}} \bar{V}_{\alpha,k}^{(c)} U_{k,i}^t \right)^2 = \\
 &= \frac{1}{n} \sum_{k=m+1}^n \sum_{l=m+1}^n \sqrt{\lambda_k^{(c)}} \sqrt{\lambda_l^{(c)}} \sum_{\alpha=1}^M \sum_{i=1}^n U_{i,l} \bar{V}_{l,\alpha}^{(c)} {}^t \bar{V}_{\alpha,k}^{(c)} U_{k,i}^t = \\
 &= \frac{1}{n} \sum_{k=m+1}^n \sum_{l=m+1}^n \sqrt{\lambda_k^{(c)}} \sqrt{\lambda_l^{(c)}} \sum_{i=1}^n (U_{k,i}^t U_{i,l}) \sum_{\alpha=1}^M (\bar{V}_{l,\alpha}^{(c)} {}^t \bar{V}_{\alpha,k}^{(c)}) = \\
 &= \frac{1}{n} \sum_{k=m+1}^n \sum_{l=m+1}^n \sqrt{\lambda_k^{(c)}} \sqrt{\lambda_l^{(c)}} \delta_{k,l} \delta_{k,l} = \frac{1}{n} \sum_{k=m+1}^n \lambda_k^{(c)}.
 \end{aligned}$$

Використовуючи (2.15), отримуємо похибку

$$\frac{\sum_{i=m+1}^n \lambda_i^{(c)}}{\sum_{i=1}^n \lambda_i^{(c)}} = 0,05. \quad (2.22)$$

Звідси випливає, що (2.21) є дуже гарним наближенням.

2.4 Метод головних компонент для незцентрованої матриці даних

У цьому розділі ми розвиваємо застосування методу головних компонент для незцентрованої матриці даних. Такий підхід виявився дуже продуктивним для модифікації генетичного алгоритму з метою прискорення процедури знаходження околу Парето-фронту як розв'язку задачі оптимізації

роботи системи автоматизованого проектування і розрахунку GeantV. Матриця даних складається з параметрів, важливих для визначення ефективності застосування системи автоматизованого проектування і розрахунку GeantV для оброблення експериментальних даних, тоді як цільові функції є функціями придатності, що їх використовують для оцінки ефективності симуляцій із застосуванням системи автоматизованого проектування і розрахунку GeantV.

За аналогією з методом головних компонент, розглянутим в попередньому розділі для центрованої матриці даних $\hat{X}^{(c)}$, ми будемо розвивати застосування цього методу для незцентрованої матриці даних $\hat{X}^{(u)}$.

У цьому випадку замість матриці коваріації

$$\hat{\Sigma} = \{\Sigma_{i,j}\} = \frac{1}{M} X_{i,\alpha}^{(c)} {}^t X_{\alpha,j}^{(c)} \quad (2.23)$$

зручно використовувати матрицю незцентрованих других моментів,

$$\hat{T} = \frac{1}{M} \hat{X}^{(u)} {}^t \cdot \hat{X}^{(u)} = \{T_{i,j}\} = \frac{1}{M} X_{i,\alpha}^{(u)} {}^t X_{\alpha,j}^{(u)}. \quad (2.24)$$

Визначимо \vec{w}_j як власні вектори матриці незцентрованих других моментів з власними значеннями $\lambda_j^{(u)}$

$$\hat{T} \cdot \vec{w}_j = \lambda_j^{(u)} \vec{w}_j, \quad 1 \leq j \leq n,$$

які задовольняють умову ортогональності

$$\vec{w}_i {}^t \cdot \vec{w}_j = \delta_{i,j}, \quad 1 \leq i, j \leq n,$$

тоді виконується співвідношення

$$\vec{w}_j^t \cdot \hat{T} \cdot \vec{w}_j = \lambda_j^{(u)}. \quad (2.25)$$

Визначимо матрицю $W_{i,j} = \vec{w}_j = (w_j)_i$ власних векторів матриці незцентрованих других моментів, яка задовольняє умову ортогональності

$$\widehat{W}^t \cdot \widehat{W} = \widehat{I}.$$

Із співвідношення (2.25) одержуємо

$$\widehat{W}^t \cdot \hat{T} \cdot \widehat{W} = \widehat{\Lambda}^{(u)}, \quad \Lambda_{i,j}^{(u)} = \lambda_i^{(u)} \delta_{i,j}. \quad (2.26)$$

Далі визначимо $\vec{v}_j^{(u)}$ як j -ту незцентровану головну компоненту аналогічно до (2.5)

$$(v_j^{(u)})_\alpha = X_{\alpha,i}^{(u)} W_{i,j} = X_{\alpha,i}^{(u)} (\vec{w}_j)_i. \quad (2.27)$$

Для дисперсії j -тої незцентрованої головної компоненти одержуємо

$$\begin{aligned} \text{Var}(\vec{v}_j^{(u)}) &= \sigma_{v,(u)}^2 = \frac{1}{M} \sum_{\alpha=1}^M \left[\sum_{i=1}^n (X_{\alpha,i}^{(u)} - \mu_i) W_{i,j} \right]^2 = \\ &= \vec{w}_j^t \cdot \hat{T} \cdot \vec{w}_j - (\vec{\mu}^t \cdot \vec{w}_j)^2 = \\ &= (\vec{\mu})^2 \left(\frac{\lambda_j^{(u)}}{(\vec{\mu})^2} - \cos^2(\vec{\mu}, \vec{w}_j) \right). \end{aligned}$$

Звідси виконується умова

$$\text{Var}(\vec{v}_j^{(u)}) > 0, \quad \text{if} \quad \lambda_j^{(u)} > (\bar{\mu})^2,$$

де

$$\mu_j = \frac{1}{M} \sum_{\alpha=1}^M X_{\alpha,j}^{(u)}, \quad 1 \leq j \leq n.$$

Для недіагональних елементів матриці коваріації для вектора незцентрованих головних компонент [68] подібним чином отримуємо

$$\text{Cov}(\vec{v}_i^{(u)}, \vec{v}_j^{(u)}) = -(\bar{\mu})^2 \cos(\vec{\mu}, \vec{w}_i) \cos(\vec{\mu}, \vec{w}_j),$$

а для косинуса кута між i -тою центрованою головною компонентою та j -тою незцентрованою головною компонентою маємо

$$\cos(\vec{v}_i^{(c)}, \vec{v}_j^{(u)}) = \sqrt{\frac{\lambda_i^{(c)}}{\lambda_i^{(u)}}} \cos(\vec{u}_i, \vec{w}_j). \quad (2.28)$$

Звідси очевидно, що для незцентрованої матриці даних вже не існує простого і ясного співвідношення між власними значеннями $\lambda_j^{(u)}$ та дисперсією j -тої незцентрованої головної компоненти $\sigma_{v,(u)}^2$, як це було у випадку центрованої матриці даних. Виходить, що аналіз в рамках наближення, регульованого «параметром головних компонент» є громіздким, і тому треба використовувати наближення, регульоване «параметром власних значень». Для цього наближення зручно використовувати сингулярне представлення для незцентрованої матриці даних, яке, як виявляється, є зручним для модифікації генетичних алгоритмів.

2.4.1 Підхід, що базується на сингулярному представленні матриці даних

Визначимо матрицю незцентрованих головних компонент $V_{\alpha,j}^{(u)} = (v_j^{(u)})_{\alpha}$, яка з (2.27) задовольняє співвідношенню

$$V_{\alpha,j}^{(u)} = X_{\alpha,i}^{(u)} W_{i,j}, \quad 1 \leq \alpha \leq M, \quad (2.29)$$

причому

$$V_{i,\alpha}^{(u)} {}^t V_{\alpha,j}^{(u)} = M \Lambda_{i,j}^{(u)} = M \lambda_i^{(u)} \delta_{i,j}. \quad (2.30)$$

Визначимо ортогональну матрицю головних компонент $\bar{V}_{\alpha,j}^{(u)}$

$$V_{\alpha,j}^{(u)} = \sqrt{M} \bar{V}_{\alpha,i}^{(u)} \Lambda_{i,j}^{(u) 1/2}, \quad (2.31)$$

де

$$\Lambda_{i,j}^{(u) 1/2} = \left[\lambda_i^{(u)} \right]^{\frac{1}{2}} \delta_{i,j},$$

Використовуючи (2.30) і (2.31), одержуємо співвідношення ортогональності

$$\bar{V}_{i,\alpha}^{(u)} {}^t \bar{V}_{\alpha,j}^{(u)} = \delta_{i,j}.$$

За допомогою співвідношень (2.31), (2.29) та (2.25), отримуємо

$$\bar{V}_{i,\alpha}^{(u) \text{ } t} X_{\alpha,k}^{(u)} X_{k,\beta}^{(u) \text{ } t} \bar{V}_{\beta,j}^{(u)} = \Lambda_{ij}^{(u)} = \lambda_i^{(u)} \delta_{ij}.$$

Звідси маємо, що $\bar{V}_{\alpha,j}^{(u)}$ є матрицею власних векторів $(\bar{v}_j^{(u)})_\alpha$ для матриці Грама $\hat{K}^{(u)} = \hat{X}^{(u)} \cdot X^{(u) \text{ } t}$ розміру $M \times M$

$$K_{\alpha,\beta}^{(u)} (\bar{v}_j^{(u)})_\beta = X_{\alpha,k}^{(u)} X_{k,\beta}^{(u) \text{ } t} (\bar{v}_j^{(u)})_\beta = \lambda_j^{(u)} (\bar{v}_j^{(u)})_\alpha$$

або в матричній формі

$$K_{\alpha,\beta}^{(u)} \bar{V}_{\beta,i}^{(u)} = \bar{V}_{\alpha,j}^{(u)} \Lambda_{j,i}^{(u)}. \quad (2.32)$$

Із співвідношення (2.29) одержуємо представлення для незцентрованої матриці даних у вигляді

$$X_{\alpha,i}^{(u)} = V_{\alpha,j}^{(u)} W_{j,i}^{t}, \quad 1 \leq \alpha \leq M, \quad (2.33)$$

за допомогою якого одержимо сингулярне представлення для незцентрованої матриці даних

$$X_{\alpha,i}^{(u)} = \sqrt{M} \bar{V}_{\alpha,k}^{(u)} \Lambda_{k,j}^{(u) \text{ } 1/2} W_{j,i}^{t}. \quad (2.34)$$

Припустимо, що матриця незцентрованих других моментів \hat{T} має p найбільших власних значень таких, що виконується співвідношення

$$\frac{\sum_{i=1}^p \lambda_i^{(u)}}{\sum_{i=1}^n \lambda_i^{(u)}} = 0,95. \quad (2.35)$$

Тоді, використовуючи наближення, регульоване «параметром власних значень» , одержуємо наближену вихідну матрицю даних $\tilde{X}_{\alpha,j}^{(u)}$ рангу p

$$\begin{aligned}\tilde{X}_{\alpha,i}^{(u)} &= \sqrt{M} \bar{V}_{\alpha,k}^{(u)} \tilde{\Lambda}_{k,j}^{(u) 1/2} W_{j,i}^t = \\ &= \sqrt{M} \left(\left[\lambda_1^{(u)} \right]^{\frac{1}{2}} \bar{V}_{\alpha,1}^{(u)} W_{1,i}^t + \dots + \left[\lambda_p^{(u)} \right]^{\frac{1}{2}} \bar{V}_{\alpha,p}^{(u)} W_{p,i}^t \right),\end{aligned}\quad (2.36)$$

де матриця власних значень $\tilde{\Lambda}_{i,j}^{(u)}$ має ранг p ($\lambda_{p+1}^{(u)} = \lambda_{p+2}^{(u)} = \dots = \lambda_n^{(u)} = 0$).

Обчислимо середньоквадратичну похибку η_p для цього наближення

$$\begin{aligned}\eta_p &= \frac{1}{Mn} \sum_{\alpha=1}^M \sum_{i=1}^n (X_{\alpha,i}^{(u)} - \tilde{X}_{\alpha,i}^{(u)})^2 = \\ &= \frac{1}{Mn} \sum_{\alpha=1}^M \sum_{i=1}^n \left(\sqrt{M} \sum_{k=p+1}^n \sqrt{\lambda_k^{(u)}} \bar{V}_{\alpha,k}^{(u)} W_{k,i}^t \right)^2 = \\ &= \frac{1}{n} \sum_{k=p+1}^n \sum_{l=p+1}^n \sqrt{\lambda_k^{(u)}} \sqrt{\lambda_l^{(u)}} \sum_{\alpha=1}^M \sum_{i=1}^n W_{i,l} \bar{V}_{l,\alpha}^{(u) t} \bar{V}_{\alpha,k}^{(u)} W_{k,i}^t = \\ &= \frac{1}{n} \sum_{k=p+1}^n \sum_{l=p+1}^n \sqrt{\lambda_k^{(u)}} \sqrt{\lambda_l^{(u)}} \sum_{i=1}^n (W_{k,i}^t W_{i,l}) \sum_{\alpha=1}^M (\bar{V}_{l,\alpha}^{(u) t} \bar{V}_{\alpha,k}^{(u)}) = \\ &= \frac{1}{n} \sum_{k=p+1}^n \sum_{l=p+1}^n \sqrt{\lambda_k^{(u)}} \sqrt{\lambda_l^{(u)}} \delta_{k,l} \delta_{k,l} = \frac{1}{n} \sum_{k=p+1}^n \lambda_k^{(u)}.\end{aligned}\quad (2.37)$$

Використовуючи (2.34) одержуємо оцінку

$$\frac{\sum_{i=p+1}^n \lambda_i^{(u)}}{\sum_{i=1}^n \lambda_i^{(u)}} = 0,05. \quad (2.38)$$

Звідси випливає, що (2.36) є непоганим наближенням.

2.5 Оператор редукції цільових параметрів в генетичних алгоритмах для багатоблочних паралельних застосувань

Ми сформулювали наближення незцентрованої матриці даних $X_{\alpha,i}^{(u)}$ за допомогою матриці $\tilde{X}_{\alpha,i}^{(u)}$, яка має нижчий ранг p , ніж вхідна матриця. Для цього використали наближення, регульоване «параметром власних значень». При цьому оцінка похибки наближення визначається найменшими власними значеннями матриці незцентрованих других моментів \hat{T} . Наприклад, коли вона має $(n - p)$ найменших власних значень $\lambda_k^{(u)} \ll 1, p + 1 \leq j \leq n$, то наближена матриця має ранг p .

Для наближення, регульованого «параметром власних значень» («eigenvalue control parameter») ми одержуємо перетворення

$$\tilde{X}_{\alpha,i}^{(u)} = \hat{R} \cdot X_{\alpha,i}^{(u)}, \quad (2.39)$$

де R -операція виконує послідовність перетворень з матрицею даних $X_{\alpha,i}^{(u)}$, а саме:

а) знаходження сингулярного представлення (2.33) для матриці даних

$$X_{\alpha,i}^{(u)} = \sqrt{M} \bar{V}_{\alpha,i}^{(u)} \Lambda_{i,j}^{(u) 1/2} W_{j,i}^t;$$

б) розташування власних значень $\lambda_i^{(u)}$ за їхньою величиною;

с) перетворення вхідної матриці власних значень $\Lambda_{ij}^{(u)}$ до вихідної матриці власних значень $\tilde{\Lambda}_{ij}^{(u)}$ рангу p , в яких покладено $\lambda_{p+1}^{(u)} = \lambda_{p+2}^{(u)} = \dots = \lambda_n^{(u)} = 0$, якщо виконується $\lambda_k^{(u)} \ll 1$, $p + 1 \leq j \leq n$.

В результаті одержуємо наближену матрицю даних

$$\tilde{X}_{\alpha,i}^{(u)} = \sqrt{M} \left([\lambda_1^{(u)}]^{\frac{1}{2}} \bar{V}_{\alpha,1}^{(u)} W_{1,i}^t + \dots + [\lambda_p^{(u)}]^{\frac{1}{2}} \bar{V}_{\alpha,p}^{(u)} W_{p,i}^t \right). \quad (2.40)$$

При застосуванні генетичних алгоритмів до незцентрованої матриці даних через дію повного генетичного оператора \hat{G} (1.11) маємо:

$$\hat{G}: \hat{X}^{(u)} \rightarrow \hat{X}'^{(u)}, \quad \hat{X}'^{(u)} = \hat{G} \cdot \hat{X}^{(u)} = (\hat{C} \circ \hat{U} \circ \hat{F}) \cdot \hat{X}^{(u)}, \quad (2.41)$$

Тепер застосування R -операції над $X_{\alpha,i}^{(u)}$ призводить до модифікації дії генетичного алгоритму на матриці даних і в результаті одержуємо

$$X'_{\alpha,i}{}^{(u)} = \hat{G} \cdot \tilde{X}_{\alpha,i}^{(u)}, \quad (2.42)$$

Поза тим вихідну матрицю $X'_{\alpha,i}{}^{(u)}$ можна записати в сингулярному представленні

$$X'_{\alpha,i}{}^{(u)} = \sqrt{M} \bar{V}'_{\alpha,k}{}^{(u)} \Lambda'_{k,j}{}^{(u) 1/2} W'_{j,i}{}^t. \quad (2.43)$$

і знову застосувати R -операцію.

Включення оператора \hat{R} (R -операція) в стандартний генетичний

алгоритм визначає дію модифікованого генетичного оператора на вектори даних

$$\hat{G}_R(\vec{x}) = (\hat{C} \circ \hat{U} \circ \hat{F} \circ \hat{R}) (\vec{x}). \quad (2.44)$$

і дає змогу перевірити чутливість параметрів оптимізації до цієї модифікації генетичного алгоритму та знайти можливі кореляції між параметрами.

Для застосування методу незцентрованих головних компонент в прикладних задачах, наприклад в генетичних алгоритмах, дуже зручно використовувати метод проєкцій. Розглянемо в рамках цього методу оператор \hat{P} , який проєктує матрицю даних \hat{X} на підпростір, натягнений на головні осі з найбільшими власними значеннями $\lambda_i^{(u)}$.

Для визначення оператора \hat{P} розглянемо матрицю $\widetilde{W}_{i,k'} = \{\vec{w}_{k'}\} = (\mathbf{w}_{k'})_i$ ($1 \leq k' \leq p$) розміру $n \times p$. Ця матриця складається з перших p найбільших власних векторів $\vec{w}_{k'}$ (2.25). Проектор $\hat{P}^{(1,p)}$ визначається наступним чином

$$\hat{P}_{ij}^{(1,p)} = \widetilde{W}_{i,k'} \widetilde{W}_{k',j}^t, \quad \hat{P}^{(1,p)} \cdot \hat{P}^{(1,p)} = \hat{P}^{(1,p)}.$$

Легко показати, що наближену матрицю даних $\tilde{X}_{\alpha,i}$, визначену в (2.36), можна одержати, діючи цим проектором праворуч на вхідну матрицю даних \hat{X} , тоді маємо

$$\tilde{X}_{\alpha,j} = X_{\alpha,i} \hat{P}_{ij}^{(1,p)} = \lambda_1^{(u)} V_{\alpha,1}^{(u)} W_{1,j}^t + \dots + \lambda_p^{(u)} V_{\alpha,p}^{(u)} W_{p,j}^t.$$

Результати застосування цього проектору в тестових багатокритеріальних задачах оптимізації показують, що власні вектори із старшими власними

значеннями визначають підпростір розв'язків багатокритеріальної задачі оптимізації, в якому лежить Парето-фронт. Застосовуючи метод ітерацій для дії \hat{P} на незцентровану матрицю даних в багатокритеріальній задачі оптимізації, ми можемо одержати прискорення збіжності до простору оптимальних розв'язків.

Включення проекційного оператора \hat{P} в модифікований генетичний оператор \hat{G}_R є дуже зручним для застосування модифікованих генетичних алгоритмів в задачах багатокритеріальної оптимізації симуляцій транспорту елементарних частинок за допомогою системи автоматизованого проектування і розрахунку GeantV, який включає роботу з високопродуктивними обчисленнями у ФВЕ, що використовують велику кількість паралельних додатків. В цьому випадку проекційний оператор \hat{P} можна записати у блочному представленні, де кожний блок діє незалежно у відповідному підпросторі даних, що опрацьовуються відповідним паралельним додатком.

Висновки до розділу 2

1. Сформульовано математичну модель методу нецентрованих головних компонент, отримано формулу для оцінки наближень при застосуванні методу для аналізу даних. Оцінка похибки наближення (2.37) показує, що процедура редукції матриці даних на основі виділення власних векторів матриці незцентрованих других моментів \hat{T} для яких вона має найменші власні значення, є коректною. Це дає змогу припустити, що використання методу головних компонент для незцентрованої матриці даних для ГА може істотно прискорити процеси пошуку Парето-фронту для задачі багатокритеріальної оптимізації.

2. Сформульована модифікація генетичного алгоритму завдяки

введенню до стандартного набору операторів ГА (селекція, мутація, кросовер) нового оператора \hat{R} , що визначається через незцентрований метод головних компонент (R -операція), і який діє на популяціях ГА. Використання методу головних компонент дає змогу перевірити чутливість параметрів генетичного алгоритму та можливу кореляцію між параметрами.

3. У випадку простого генетичного алгоритму ПГА [102], швидкість збіжності генетичного алгоритму залежить від власних значень модифікованого повного генетичного оператора \hat{G}_R , що йдуть слідом за старшим власним значенням. Другим важливим спостереженням є те, що власні вектори з найбільшими власними значеннями можуть визначати підпростір задачі багатокритеріальної оптимізації, в якому лежить Парето-фронт. Використовуючи ітераційну процедуру застосування \hat{G}_R для незцентрованої матриці даних, ми сподіваємось швидко наблизитися до оптимального підпростору, в якому лежить Парето-фронт.

Результати досліджень за другим розділом опубліковані в статтях:

1. Shadura, O., Petrenko A. and Svistunov S. Multivariate convergence-targeted operator for the genetic. Системні дослідження та інформаційні технології : міжнародний науково-технічний журнал., 2017, № 1., стр. 126–140.

2. Shadura, O., Carminati, F. and Petrenko A. Performance Optimization of Physics Simulations Through Genetic Algorithms, Journal of Computer Science, 2019, v.15, n.1, p. 57-66, DOI: 10.3844/jcssp.2019.57.66.

РОЗДІЛ 3

РЕЗУЛЬТАТИ ЗАСТОСУВАННЯ НГК-ОПЕРАТОРА ДЛЯ ЕТАЛОННИХ ТЕСТІВ БАГАТОЦІЛЬОВИХ ЗАДАЧ

3.1 Стандартні методи перевірки нових алгоритмів та набір тестів DTLZ

Тести є одним з поширених способів перевірки коректності роботи генетичних алгоритмів, вони дають змогу порівняти ефективність різних генетичних алгоритмів, їх вихідний розподіл, статистику і провести аналіз роботи операторів в генетичних алгоритмах та загальний аналіз продуктивності стохастичного алгоритму. Простими прикладами таких тестів є стандартний набір DTLZ-тестів (Deb, Thiele, Laumanns, Zitzler) для багатоцільових задач оптимізації [39]. Інші популярні тести можуть бути знайдені для стандартного набору ZDT (Zitzler-Deb-Thiele) за посиланнями [121], для стандартного набору LZ09 – за посиланнями [76], та для стандартного набору WFG – за посиланнями [64].

Тести DTLZ сконструйовані таким чином, що багатоцільові задачі оптимізації можуть масштабуватися до будь-якої кількості функцій. Крім того, всі DTLZ1-DTLZ6 масштабуються стосовно кількості параметрів, але мають фіксовану кількість вихідних параметрів $M - 1$, де M – кількість цільових функцій. Ці вихідні параметри є взаємопов'язаними, оскільки спроба оптимізувати їх за одним параметром одночасно (лише за один прохід) не буде визначати всі глобальні оптимуми. Відзначимо деякі властивості тестів: цільові функції в DTLZ1-DTLZ4 мають кілька глобальних оптимумів. DTLZ5 та DTLZ6 є тестовими задачами з виродженими оптимальними фронтами Парето, де оптимальні fronti Парето – це дуги, вкладені в M -розмірний об'єкт.

При застосуваннях багатоцільових генетичних алгоритмів до задач багатокритеріальної оптимізації їх ефективність перевіряється рішенням двох задач: а) спроможністю еволюційного алгоритму давати збіжність до оптимального фронту Парето («задача збіжності») та б) давати гарний розподіл оптимальних розв'язків по всьому Парето-фронті («задача поширення»).

Для перевірки ефективності алгоритму для рішення задачі збіжності, тестова задача повинна мати перепони для еволюції системи, наприклад, у вигляді великої кількості локальних оптимальних фронт. Для перевірки ефективності рішення задачі поширення, тестова задача повинна мати області ущільнення оптимальних розв'язків на Парето-фронті, наприклад, внаслідок існування локальних Парето-оптимальних атракторів на цьому фронті. Цього можна досягнути роблячи оптимальні Парето-фронти неопуклими, дискретними й такими, що мають змінну щільність оптимальних рішень вздовж фронту.

Набір тестів DTLZ дозволяє досліджувати ефективність методів рішення багатокритеріальних задач оптимізації контрольованим чином, з відомими характеристиками, кількістю параметрів та знанням оптимального фронту Парето для кожного тесту.

Для знаходження оптимальних значень параметрів при оптимізації симуляцій GeantV ми розглядаємо модель GeantV як задачу «чорної скриньки» зі скінченним набором фітнес функцій. З огляду на це, чим більше ми знайдемо оптимальних значень параметрів в околі Парето-оптимального фронту тим простіше буде реалізувати такі оптимальні рішення для обчислювальної інфраструктури що використовується для поточної обробки даних за допомогою GeantV. Тоді для перевірки ефективності застосованого генетичного алгоритму NSGA-II з включенням НГК-оператора для оптимізації роботи GeantV необхідно використовувати такі тестові задачі які дозволяють перевірити алгоритм на отримання широкого розподілу

оптимальних розв'язків в околі всього Парето-фронту («задача поширення»). Серед тестів DTLZ для перевірки ефективності рішення задачі поширення зручними є тести DTLZ2 та DTLZ4.

З цієї точки зору ми обмежуємось перевіркою ефективності запропонованої нами модифікації генетичних алгоритмів за умови включення в них НГК-оператора на основі тестових задач DTLZ2 та DTLZ4 та порівняння ефективності цієї модифікації з роботою генетичних алгоритмів без НГК-оператора.

В цьому розділі ми приводимо результати тестування модифікації генетичних алгоритмів за умови включення в них НГК-оператора та порівняння ефективності цієї модифікації з роботою генетичних алгоритмів без НГК-оператора. Давайме коротко опишемо кожен з DTLZ тестових задач, включаючи DTLZ2 та DTLZ4 [90].

Тестова задача DTLZ1. Ця тестова задача є найпростішою М-цільовою задачею оптимізації з лінійним оптимальним Парето-фронтом:

$$\begin{aligned}
 \text{minimize } f_1(\vec{x}) &= 0.5(1 + g(\vec{z})) \cdot x_1 \cdot x_2 \cdot \dots \cdot x_{M-1}, \\
 \text{minimize } f_2(\vec{x}) &= 0.5(1 + g(\vec{z})) \cdot x_1 \cdot x_2 \cdot \dots \cdot x_{M-2} \cdot (1 - x_{M-1}), \\
 &\dots \\
 \text{minimize } f_m(\vec{x}) &= 0.5(1 + g(\vec{z})) \cdot \prod_{l=1}^{M-m} x_l \cdot (1 - x_{M-m+1}), \\
 &\dots \\
 \text{minimize } f_M(\vec{x}) &= 0.5(1 + g(\vec{z})) \cdot (1 - x_1),
 \end{aligned} \tag{3.1}$$

де

$$g(\vec{z}) = 100 \cdot (n + \sum_{i=1}^n [(z_i - 0.5)^2 - \cos(20\pi(z_i - 0.5))]),$$

тут $\vec{x} = (x_1, x_2, \dots, x_{M-1})$, цільові функції $f_l(\vec{x})$ ($l = 1, \dots, M$) залежать від n параметрів $\vec{z} = (z_1, z_2, \dots, z_n)$, причому змінні x_l лежать в інтервалі $0 \leq$

$x_l \leq 1$, $l = 1, \dots, M - 1$ і параметри z_i – в інтервалі $0 \leq z_i \leq 1$, $i = 1, \dots, n$. Зауважимо, що ці інтервали для змінних \vec{x} та параметрів \vec{z} будуть використані також і для тестових задач DTLZ2 -DTZL4.

Парето-оптимальний розв'язок для DTLZ1 відповідає нульовим значенням параметрів $\vec{z}^* = (0, 0, \dots, 0)$ та оптимальні значення цільових функцій розташовані на лінійній гіпер-площині:

$$\sum_{l=1}^M f_l^* = 0.5.$$

В статті [90] було показано що ця багато-цільова задача також має $(11^n - 1)$ локальних оптимальних фронтів Парето, кожний з яких може притягувати точки системи що еволюціонують за допомогою багатоцільового еволюційного алгоритму.

У цьому розділі ми використовували число параметрів $\{z_1, z_2, \dots, z_n\}$, що дорівнювало $n = 5$, яке часто використовується в літературі. Для цього випадку оптимальний фронт Парето знайдений за допомогою генетичного алгоритму NSGA-II зображений на рисунку 3.1, де для зручності обмежились випадком $M = 3$.

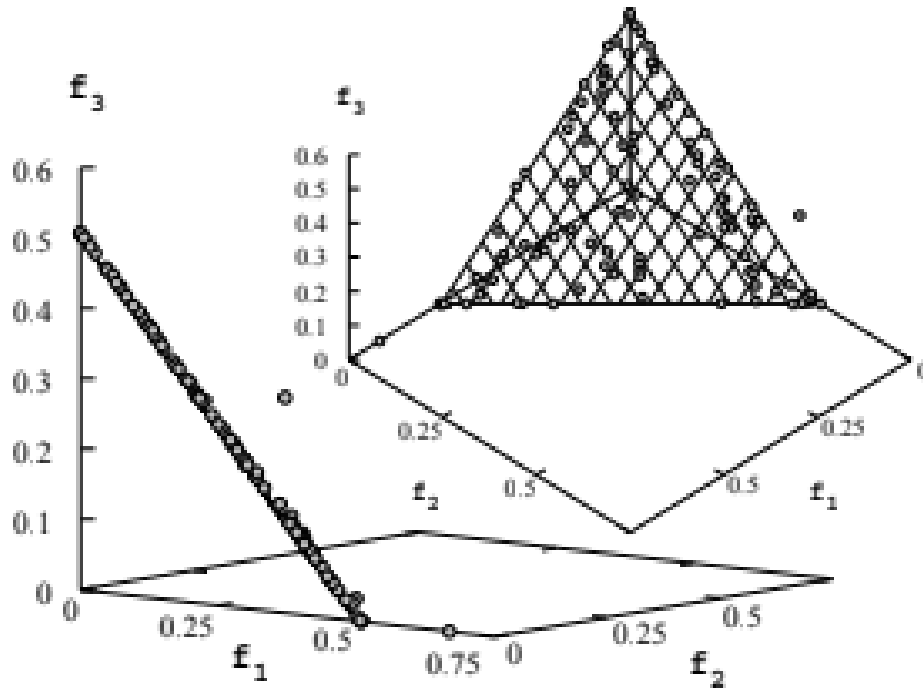


Рисунок 3.1 – Фронт Парето для тесту DTLZ1 (NSGA-II)

Тестова задача DTLZ2. Ця тестова задача є М-цільовою задачею оптимізації з випуклим оптимальним фронтом Парето:

$$\text{minimize } f_1(\vec{x}) = (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-1}\right)$$

$$\text{minimize } f_2(\vec{x}) = (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-2}\right) \cdot$$

$$\sin\left(\frac{\pi}{2} x_{M-1}\right)$$

...

$$\text{minimize } f_m(\vec{x}) = (1 + g(\vec{z})) \cdot \prod_{l=1}^{M-m} \cos\left(\frac{\pi}{2} x_l\right) \cdot$$

$$\sin\left(\frac{\pi}{2} x_{M-m+1}\right)$$

...

$$\text{minimize } f_M(\vec{x}) = (1 + g(\vec{z})) \cdot \sin\left(\frac{\pi}{2} x_1\right) \quad (3.2)$$

де

$$g(\vec{z}) = \sum_{i=1}^n (z_i - 0.5)^2.$$

Оптимальний розв'язок Парето для DTLZ2 відповідає значенням параметрів $\vec{z}^* = (0.5, 0.5, \dots, 0.5)$ та оптимальні значення цільових функцій розташовані на гіперповерхні:

$$\sum_{i=1}^M (f_i^*)^2 = 1.$$

В цьому розділі ми використовували число параметрів $n = 10$, яке часто використовується в літературі для цього випадку. Парето-оптимальний фронт знайдений за допомогою генетичного алгоритму NSGA-II зображений на рисунку 3.2. (для зручності використали $M = 3$)

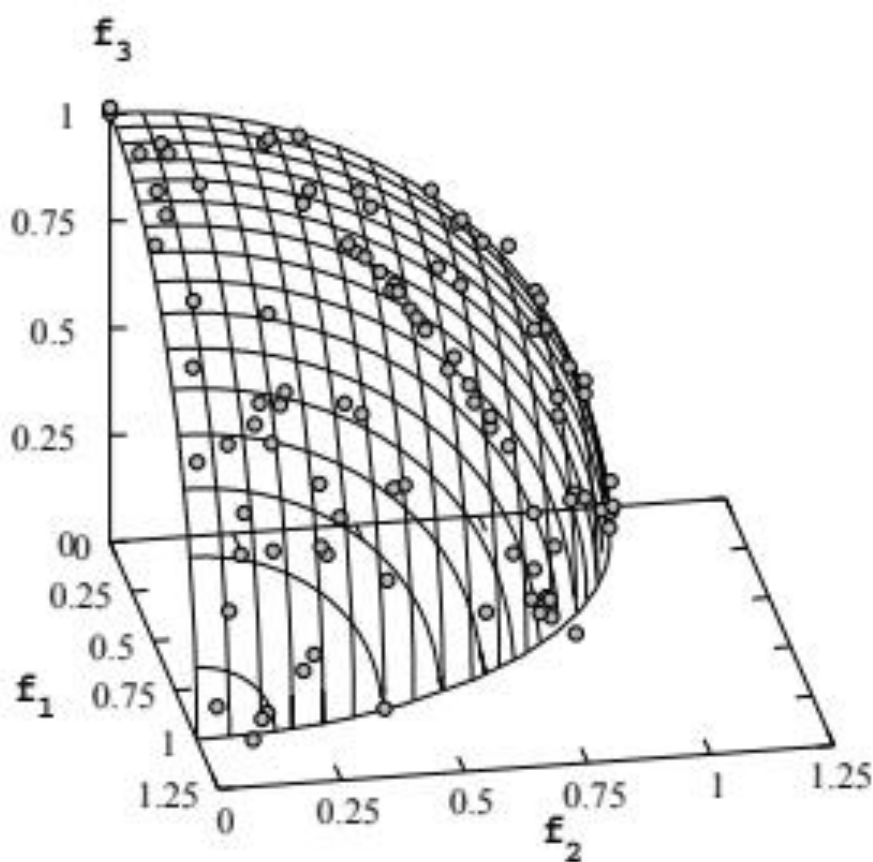


Рисунок 3.2 – Фронт Парето для тесту DTLZ2 (NSGA-II)

Тестова задача DTLZ3. Ця тестова задача є М-цільовою задачею оптимізації для перевірки ефективності генетичного алгоритму для рішення «задачі збіжності» оскільки має багато локальних Парето-оптимальних фронтів:

$$\begin{aligned}
 \text{minimize } f_1(\vec{x}) &= (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-1}\right) \\
 \text{minimize } f_2(\vec{x}) &= (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-2}\right) \cdot \\
 &\quad \sin\left(\frac{\pi}{2} x_{M-1}\right) \dots \\
 \text{minimize } f_m(\vec{x}) &= (1 + g(\vec{z})) \cdot \prod_{l=1}^{M-m} \cos\left(\frac{\pi}{2} x_l\right) \cdot \sin\left(\frac{\pi}{2} x_{M-m+1}\right) \\
 &\dots \\
 \text{minimize } f_M(\vec{x}) &= (1 + g(\vec{z})) \cdot \sin\left(\frac{\pi}{2} x_1\right)
 \end{aligned} \tag{3.3}$$

де

$$g(\vec{z}) = 100 \cdot (n + \sum_{i=1}^n [(z_i - 0.5)^2 - \cos(20\pi(z_i - 0.5))]).$$

У цьому розділі для тестової задачі DTZL3 ми викотуємо $n = 10$. Функція $g(\vec{z})$ призводить до $(3^n - 1)$ локальних оптимальних фронтів Парето та одного глобального фронту Парето. Всі локальні оптимальні fronti Парето є паралельними до глобального фронту Парето і багатоцільовий еволюційний алгоритм може привести систему до цих локальних фронтів Парето, перед тим як система попаде до глобального оптимального фронту Парето ($g^*(\vec{z}^*) = 0$). Цей глобальний фронт Парето відповідає значенням параметрів $\vec{z}^* = (0.5, 0.5, \dots, 0.5)$ та оптимальні значення цільових функцій розташовані на гіперповерхні:

$$\sum_{l=1}^M (f_l^*)^2 = 1.$$

Наступний локальний Парето-фронт відповідає $g^*(\vec{z}^*) = 1$.

Глобальний оптимальний фронт Парето знайдений за допомогою генетичного алгоритму NSGA-II, після генерації 500 поколінь в алгоритмі, зображений на рисунку 3.3, де обмежились випадком $M = 3$.

Тестова задача DTLZ4. Ця тестова задача є М-цільовою задачею оптимізації для перевірки ефективності генетичного алгоритму для рішення «задачі поширення» оскільки має локальні ущільнення оптимальних розв'язків на фронт Парето:

$$\begin{aligned}
 \text{minimize } f_1(\vec{x}) &= (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1^\alpha\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-1}^\alpha\right) \\
 \text{minimize } f_2(\vec{x}) &= (1 + g(\vec{z})) \cdot \cos\left(\frac{\pi}{2} x_1^\alpha\right) \cdot \dots \cdot \cos\left(\frac{\pi}{2} x_{M-2}^\alpha\right) \cdot \\
 &\quad \sin\left(\frac{\pi}{2} x_{M-1}^\alpha\right) \cdot \dots \\
 \text{minimize } f_m(\vec{x}) &= (1 + g(\vec{z})) \cdot \prod_{l=1}^{M-m} \cos\left(\frac{\pi}{2} x_l^\alpha\right) \cdot \\
 &\quad \sin\left(\frac{\pi}{2} x_{M-m+1}^\alpha\right) \\
 &\quad \dots \\
 \text{minimize } f_M(\vec{x}) &= (1 + g(\vec{z})) \cdot \sin\left(\frac{\pi}{2} x_1^\alpha\right)
 \end{aligned} \tag{3.4}$$

де

$$g(\vec{z}) = \sum_{i=1}^n (z_i - 0.5)^2.$$

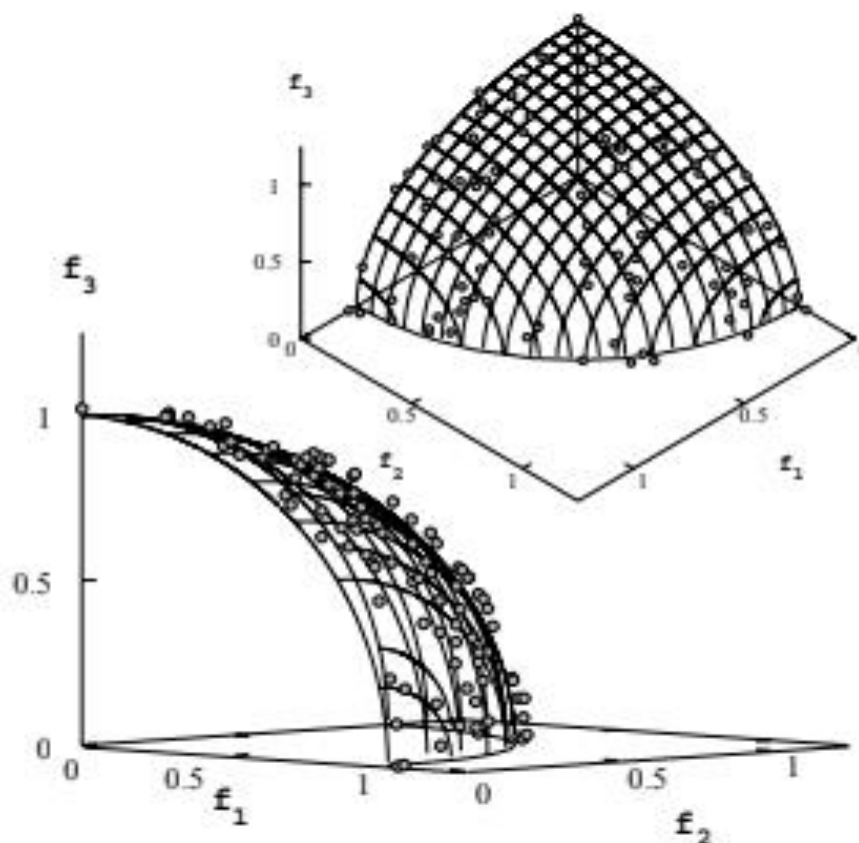


Рисунок 3.3 – Фронт Парето для тесту DTLZ3 (NSGA-II)

Як розглянуто в статті [90], виберемо значення параметра $\alpha = 100$, а параметра $n = 10$. Ця багатоцільова задача має щільну множину рішень в околі $f_M - f_1$ площини. Використовуючи генетичний алгоритм NSGA-II, цей результат отримано після 200 генерацій поколінь при $M = 3$ і показано на рисунку 3.4. У цьому підході кінцеві покоління дуже корелюють з вибором початкових поколінь і в роботі [90] було отримано три варіанти ущільнення рішень: а) оптимальні рішення знаходяться в площині $f_3 - f_1$, б) оптимальні рішення знаходяться в площині $f_1 - f_2$, в) рішення заповнюють всю Парето-оптимальну поверхню.

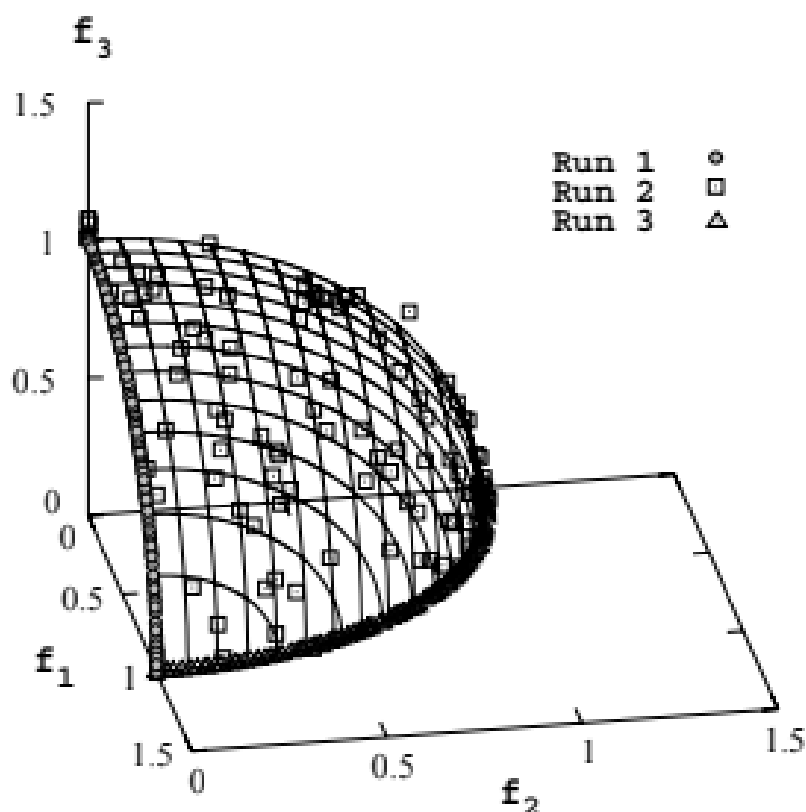


Рисунок 3.4 – Фронт Парето для тесту DTLZ4 (NSGA-II)

3.2 Вибір генетичного алгоритму

Ми використовуємо генетичний алгоритм NSGA-II [42] як основний підхід для тестування ефективності застосування НГК-оператора в задачах багатокритеріальної оптимізації. Цей алгоритм вважається одним з найбільш популярних генетичних алгоритмів і базується на ідеї швидкого сортування для домінування особин, що генеруються, і забезпечує значну швидкість збіжності для досягнення оптимального набору Парето.

Опис алгоритму NSGA-II наведено в розділі 2 та у Додатку А, а його результати показані в статті [42].

З алгоритму у Додатку А, особини в популяції групуються в набір фронтів за принципом недомінантності. Перший фронт розглядають як

абсолютно недомінантний набір у поточній популяції, на другому фронті переважають особини лише першого фронту і так далі. Для кожного окремого учасника на кожному фронті призначається оцінний ранг, що базується на номері фронту, до якого вони належать. Особи́нам у першому фронті надають рангове значення 1, особи́нам у другому – значення 2 і так далі. Окрім значення рангу, вводять новий параметр, який називається щільністю скупчення, і його розраховують для кожної особи́ни. Щільність скупчення означає, наскільки близько до своїх сусідів розташована особина. Більша середня щільність призводить до більшого розмаїття популяції. Батьків добирають з популяції за допомогою бінарних турнірів з урахуванням рангу та щільності скупчення. Між двома особи́нами добирається та, яка має менший ранг, або більшу щільність скупчення. За допомогою операторів кросовера та мутацій можна дібрати популяцію для генерування потомків. Популяцію повторно сортують за принципом недомінантності та до неї добирають лише найкращі N особи́н для створення нової популяції, де N – чисельність популяції.

Відстань від скупчення є показником того, наскільки особина близька до своїх сусідів. Якщо середнє значення відстаней витіснення є більшим, то це призводить до значного розмаїття для популяції. NSGA-III [36] має схему алгоритму, що направлена на збільшення різноманітності популяції, і побудована на ідеї вдосконаленої процедури відбору опорної точки та використання вже визначеного набору опорних точок для забезпечення різноманітності популяції.

Функція *SortByRankAndDistance* впорядковує популяцію в ієрархію недомінантного Парето-фронтів. Функція *CrowdingDistanceAssignment* обчислює середню відстань між особи́нами для кожного фронту. Функція *CrossoverAndMutation* виконує класичні кросоверні та мутаційні генетичні операції для генетичного алгоритму. Обидві функції *SelectParentsByRankAndDistance* і *SortByRankAndDistance* структурують

популяцію спочатку за рангом (порядок домінантного фронту, до якого належить розв'язок), а потім за щільністю у фронті (що розрахована за *CrowdingDistanceAssignment*).

Ми будемо використовувати алгоритм NSGA-II для порівняння алгоритмів як один з офіційно затверджених у науковій спільноті.

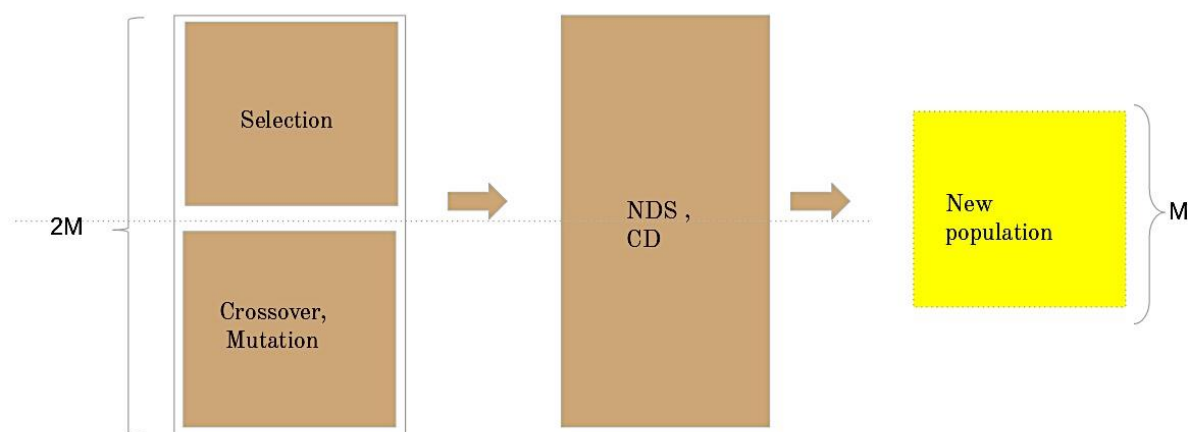


Рисунок 3.5 – Алгоритм NSGA-II

Основною проблемою генетичних алгоритмів є відсутність операторів, які прискорюють збіжність до глобальних оптимумів моделі. В наступному розділі ми протестуємо запропонований в дисертації вдосконалений алгоритм на основі комбінації підходів NSGA-II й НКГ-оператора, що був визначений у попередньому розділі.

3.3 Оптимізація схеми генетичного алгоритму

Наступним важливим кроком є перевірка того, що алгоритм є обчислювально-ефективним. У NSGA-II батьків добирають з популяції за допомогою бінарних турнірів з урахуванням рангу та параметру щільності скупчення індивідуумів (на рисунку 3.5 ця операція відмічена прямокутником з надписом Selection). Між двома особинами добирається та яка має менший ранг або більшу щільність скупчення. За допомогою операторів кросовера та

мутацій можна сформувати популяцію для генерування нащадків (на рисунку 3.5 ці операції відмічені прямокутником з надписом Crossover, Mutation). Популяцію повторно сортують за принципом не домінантності та до неї добирають лише найкращі M особин для створення нової популяції (на рисунку 3.5 ці операції відмічені прямокутником з надписом NDS, CD). В результаті ми отримуємо нову популяцію яка складається з M особин (на рисунку 3.5 цей результат відмічений прямокутником з надписом New population). Модифікація генетичного алгоритму внаслідок включення НГК-оператора (**R** -операції) на рисунках нижче відмічена прямокутником з надписом UPCA (Uncentered Principal Component Analysis)).

3.3.1 Метод NSGA-II - UPCA1 та включення НГК-оператора в генетичний алгоритм

Включення оператора \hat{R} (R -операція) в генетичний алгоритм приводить до модифікації генетичного алгоритму і дає новий повний генетичний оператор $G_R(\vec{x}) = (\hat{R} \circ \hat{C} \circ \hat{U} \circ F)(\vec{x})$.

Для оптимізації алгоритму ми маємо оцінити яку «вартість» для алгоритму привнесе новий оператор. Наша ціль зменшити вартість оцінок та отримати більшу матрицю популяцій:

$$\begin{aligned} & \text{minimize } f(N_{PreEval}), \\ & \text{minimize } f(N_{PostEval}), \\ & \text{maximize } f(space_{upca}). \end{aligned} \quad (3.5)$$

де $N_{PreEval}$ — кількість оцінок до використання НГК-оператора

(R -операції); $N_{PostEval}$ – кількість оцінок після використання НГК-оператора; $space_{upca}$ – є найбільш пріоритетним значенням для максимізації – це множина значень для використання НГК-оператора.

Найкращі особи обчислюються за допомогою недомінуючого сортування за допомогою підрахунку відстані. Загальна кількість оцінок функцій для NSGA-II:

$$N_{PreEval} = 2 * size_{population} * N_{fitness_{functions}} \quad (3.6)$$

де $size_{population}$ – розмір популяції; $N_{fitness_{functions}}$ – розмір вектору параметрів (вектора індивідууму); $f(N_{PreEval})$ – кількість оцінок для батьків та для нащадків.

Розглянемо складність однієї ітерації всього алгоритму [42]. Основні операції та їх найгірші складності такі: 1) недомінантне сортування NDS – $O(M(2N)^2)$; 2) присвоєння відстані натовпу CD (crowding distance) – $O(M(2N \log(2N)))$; 3) сортування – $O(2N \log(2N))$. Загальна складність алгоритму дорівнює $O(MN^2)$, (де M – кількість цілей, а N – чисельність популяції) [42] та регулюється невизначеною сортувальною частиною алгоритму.

Метод NSGA-II – UPCA1 включення НГК-оператора в генетичний алгоритм схематично показано на рисунку 3.6, де включення нового оператора проведено до початку операції відбору та кросовера. Це означає, що UPCA (НГК-оператор) використовується як перший крок з оригінальною популяцією і разом з відбором і мутацією створює нову популяцію розміром $3M$.

UPCA більш ефективний на великих наборах даних, що означає, що в нашому випадку на рисунку 3.6 лише одна третина покоління генетичного алгоритму наступного кроку є «безшумним». Це робить конвергенцію

генетичного алгоритму набагато більш повільною і збільшує кількість оцінок фітнес-функцій на третину:

$$\begin{aligned} N_{PreEval} &= 2 * size_{population} * N_{fitness_functions}, \\ N_{PostEval} &= size_{population} * N_{fitness_functions}, \\ space_{upca} &= size_{population} * N_{fitness_functions}. \end{aligned} \quad (3.7)$$

де $size_{population}$ – розмір популяції; $N_{fitness_functions}$ – розмір вектору оцінок; $N_{PreEval}$ – кількість оцінок до використання НГК-оператора для батьків та для нащадків; $N_{PostEval}$ – кількість оцінок після використання НГК-оператора для батьків та для нащадків; $space_{pca}$ – множина значень для використання НГК-оператора.

Слабкі результати обґрунтовуються невеликою часткою популяції, залученої до UPCA після операції недомінатного сортування NDS & CD.

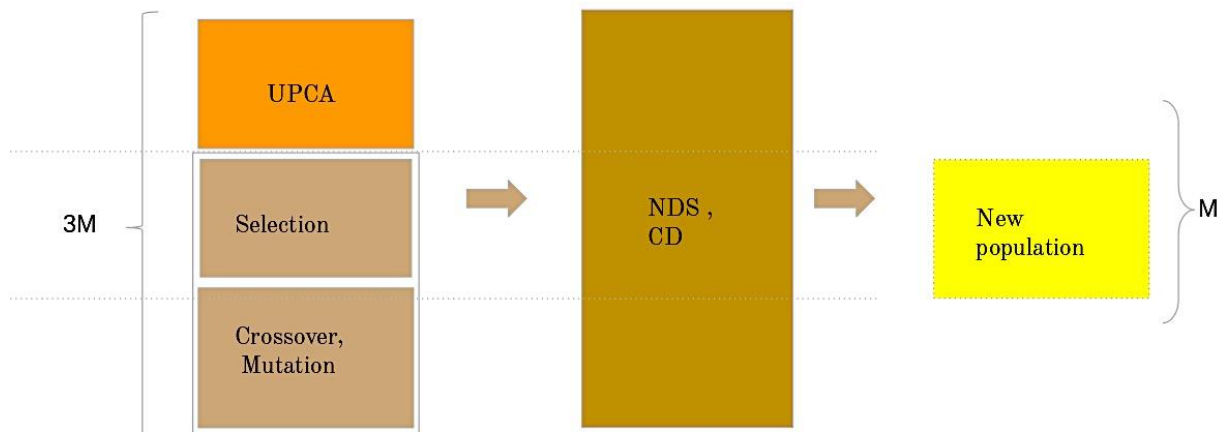


Рисунок 3.6 – Метод NSGA-II - UPCA1 та включення НГК-оператора в генетичний алгоритм

3.3.2 Метод NSGA-II - UPCA2 включення НГК-оператора в генетичний алгоритм

Цей метод включення НГК-оператора показано на рисунку 3.7, де UPCA в генетичному алгоритмі використовується в окремій матриці переходу:

$$\begin{aligned} N_{PreEval} &= 2 * size_{population} * N_{fitness_{functions}}, \\ N_{PostEval} &= 2 * size_{population} * N_{fitness_{functions}}, \\ space_{upca} &= 2 * size_{population} * N_{fitness_{functions}} \end{aligned} \quad (3.8)$$

де $size_{population}$ – розмір популяції; $N_{fitness_{functions}}$ – розмір вектору оцінок; $N_{PreEval}$ – кількість оцінок до використання НГК-оператора для батьків та для нащадків; $N_{PostEval}$ – кількість оцінок після використання НГК-оператора для батьків та для нащадків; $space_{upca}$ – множина значень для використання НГК-оператора.

Велика кількість оцінок може бути значно спрощена внаслідок простого кешування даних або створення бази значень оцінок (lookup table).

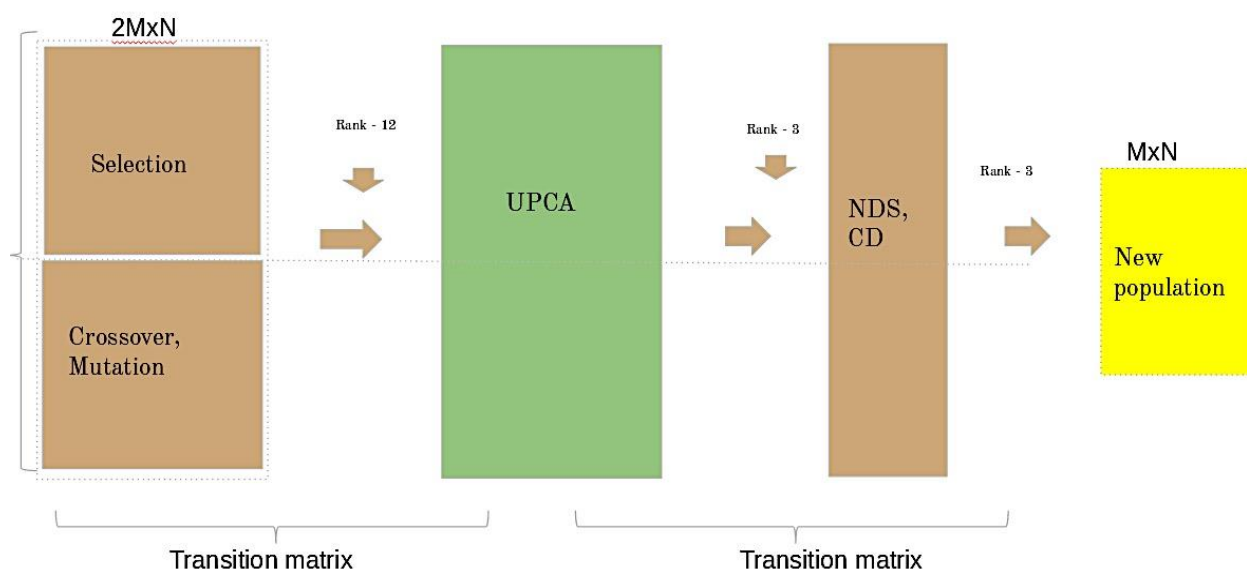


Рисунок 3.7 – Метод NSGA-II - UPCA2 та включення НГК-оператора в генетичний алгоритм

3.3.3 Метод NSGA-II - UPCA3 включення НГК-оператора в генетичний алгоритм

Цей метод включення НГК-оператора в генетичний алгоритм схематично показано на рисунку 3.8. В цьому випадку використання операцій недомінантного сортування NDS/CD два рази дає трохи кращі результати, але це не є ефективним для обчислювальних цілей, оскільки нам потрібно генерувати вдвічі більшу кількість оцінок:

$$\begin{aligned} N_{PreEval} &= 3 * size_{population} * N_{fitness_functions}; \\ N_{PostEval} &= size_{population} * N_{fitness_functions}; \\ space_{upca} &= size_{population} * N_{fitness_functions}. \end{aligned} \quad (3.9)$$

де $size_{population}$ – розмір популяції; $N_{fitness_functions}$ – розмір вектору оцінок; $N_{PreEval}$ – кількість оцінок до використання НГК-оператора для батьків та для нащадків; $N_{PostEval}$ – кількість оцінок після використання НГК-оператора для батьків та для нащадків; $space_{upca}$ – множина значень для використання НГК-оператора.

Різноманітність індивідів досягається набагато інтенсивніше, але для обчислювально-дорогих оцінок ефективні результати практично не враховуються через час оцінювання фітнес-функцій.

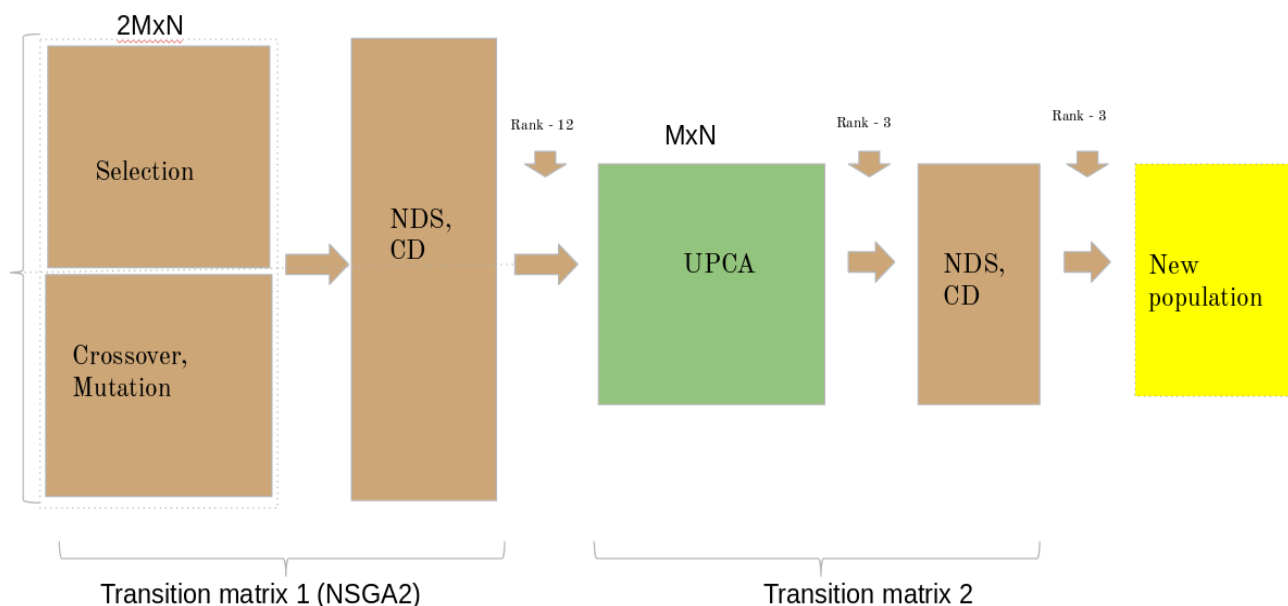


Рисунок 3.8 – Метод NSGA-II - UPCA3 та включення НГК-оператора в генетичний алгоритм

3.3.4 Метод NSGA-II - UPCA4 включення НГК-оператора в генетичний алгоритм

Метод NSGA-II - UPCA4 включення НГК-оператора схематично показано на рисунку 3.9. В цьому випадку проблема розміру популяції схожа з випадком методу NSGA-II - UPCA1 (рисунок 3.5), де слабкі результати обґрунтовуються невеликою часткою популяції для ***space_{upca}***:

$$\begin{aligned}
 N_{PreEval} &= 2 * size_{population} * N_{fitness_functions}; \\
 N_{PostEval} &= size_{population} * N_{fitness_functions}; \\
 space_{upca} &= size_{population} * N_{fitness_functions}.
 \end{aligned}
 \tag{3.10}$$

де ***size_{population}*** – розмір популяції; ***N_{fitness_functions}*** – розмір вектору

оцінок; $N_{PreEval}$ – кількість оцінок до використання НГК-оператора для батьків та для нащадків; $N_{PostEval}$ – кількість оцінок після використання НГК-оператора для батьків та для нащадків; $space_{pca}$ – множина значень для використання НГК-оператора.

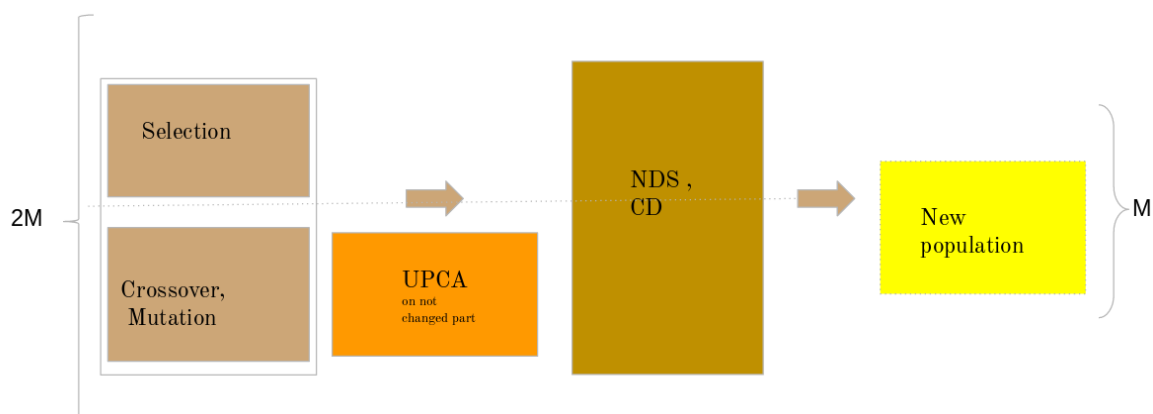


Рисунок 3.9 – Метод NSGA-II - UPCA4 включення НГК-оператора в генетичний алгоритм

3.3.5 Приклади розв'язання типових задач багатокритеріальної оптимізації

Тести DTLZ [40] являють собою набір багатокритеріальних задач, які використовуються для порівняння або валідації генетичних алгоритмів.

В цьому розділі зроблене порівняння результатів застосування алгоритмів NSGA-II [42] з включенням та без включення НГК-оператора для тестів DTLZ. Відмітимо, що ефективність алгоритму NSGA-III перевищує NSGA-II, але для простоти ми перевірили наш метод використовуючи алгоритм NSGA-II й отримані результати показують ефективність нашого підходу. Використовуючи тести DTLZ, дисертантом був проведений порівняльний аналіз ефективності використання вище приведених методів

включення НГК-оператора в генетичний алгоритм (від NSGA-II - UPCA1 до NSGA-II - UPCA4) для нецентрованих матриць даних та для центрованих матриць даних без включення НГК-оператора за допомогою оцінки середньоквадратичної похибки й показано, що метод NSGA-II - UPCA2 (рисунок 3.16) включення НГК-оператора є найбільш ефективним для збіжності модифікованого генетичного алгоритму до фронту Парето.

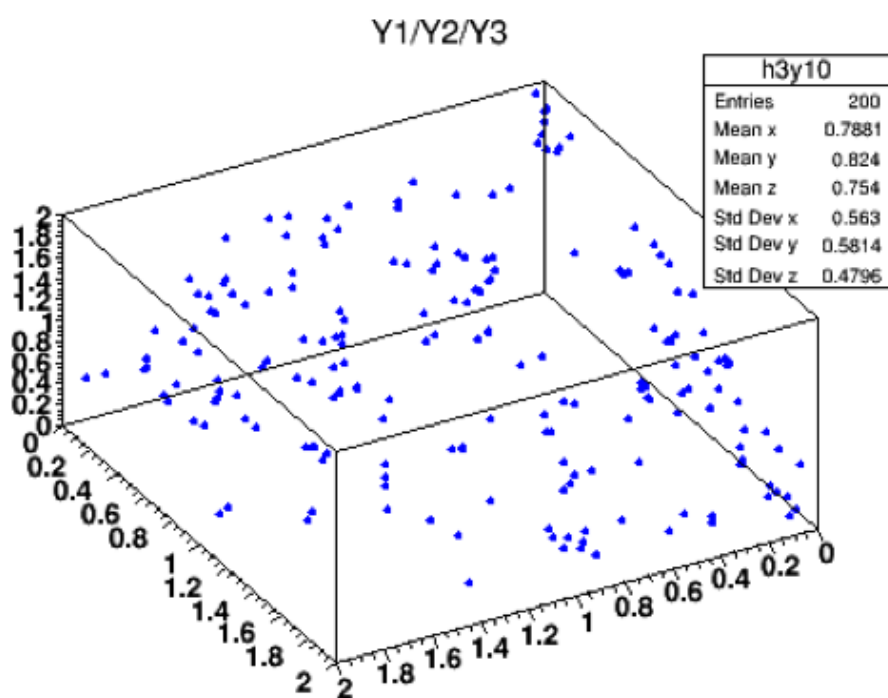


Рисунок 3.10 – Розподіл популяції для 10-го покоління для тесту DTLZ2
(алгоритм NSGA-II)

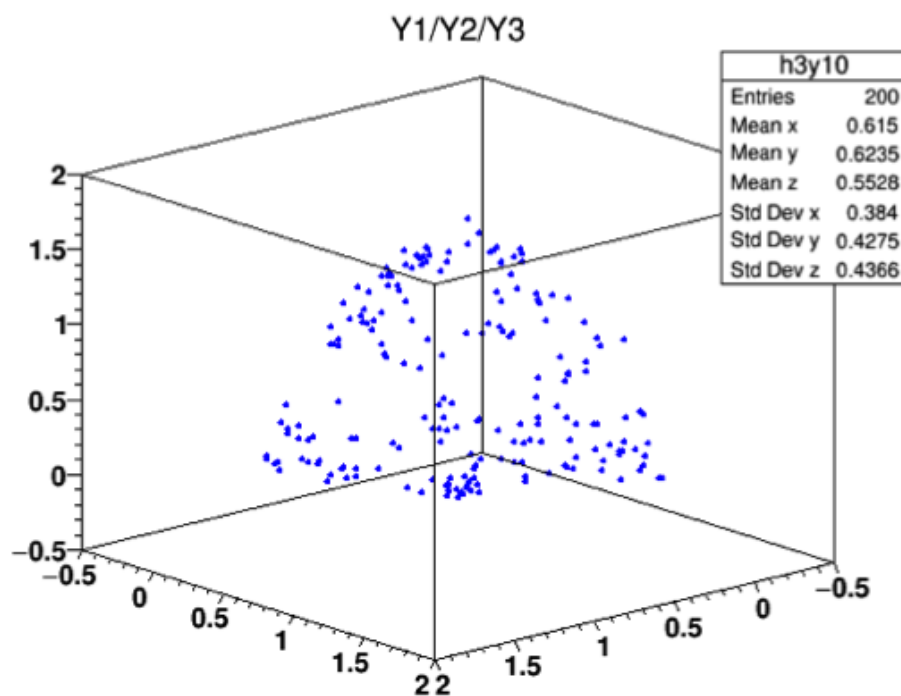


Рисунок 3.11 – Розподіл популяції для 10-го покоління для тесту DTLZ2
(метод NSGA-II - UPCA2)

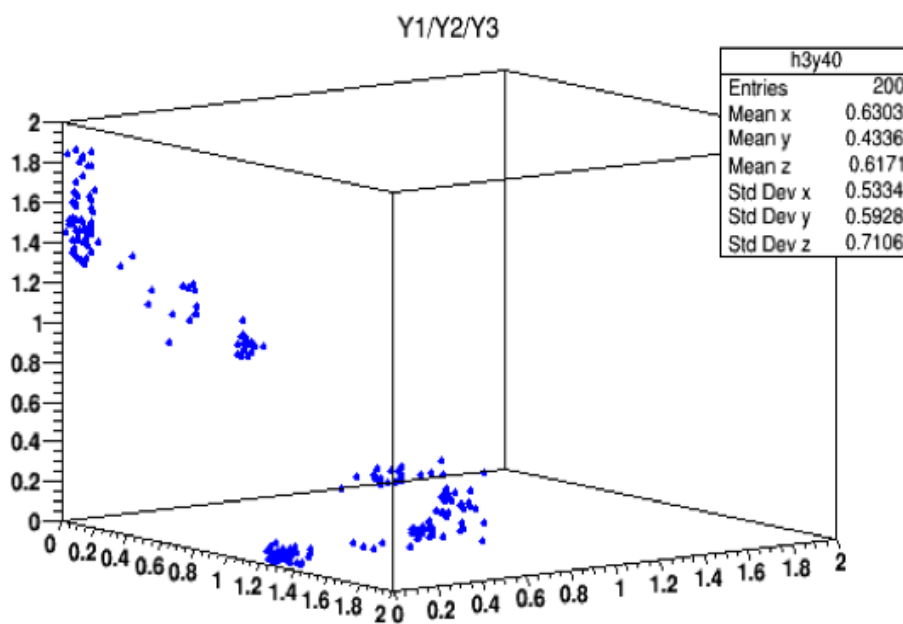


Рисунок 3.12 – Розподіл популяції для 10-го покоління для тесту DTLZ4
(алгоритм NSGA-II)

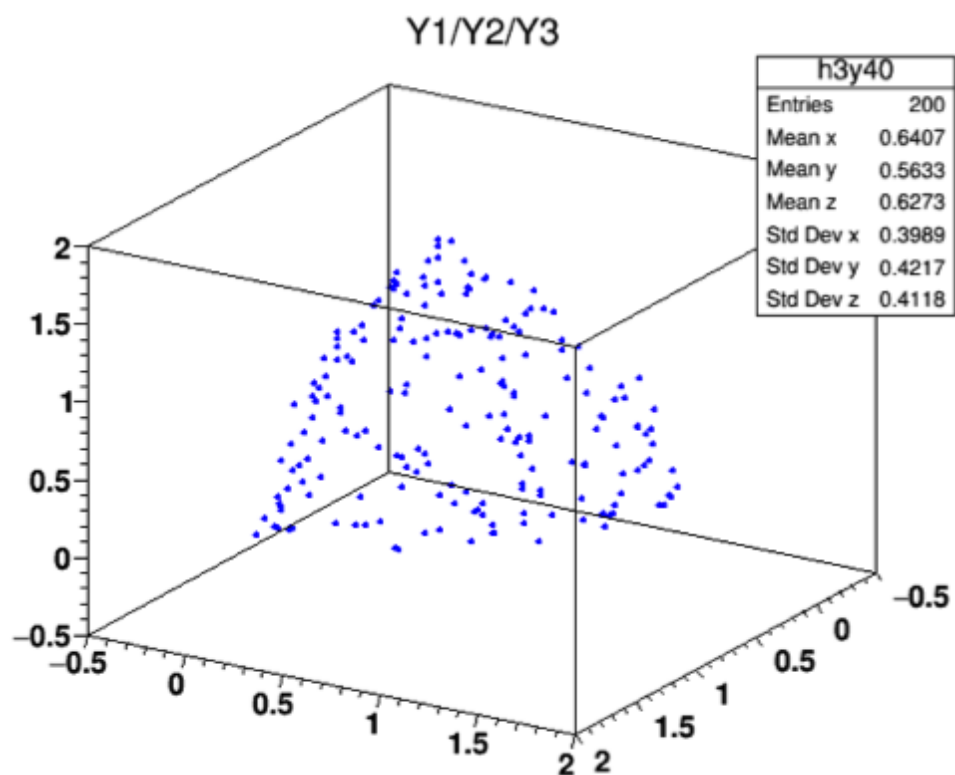


Рисунок 3.13 – Розподіл популяції для 10-го покоління для тесту DTLZ4
(метод NSGA-II - UPCA2)

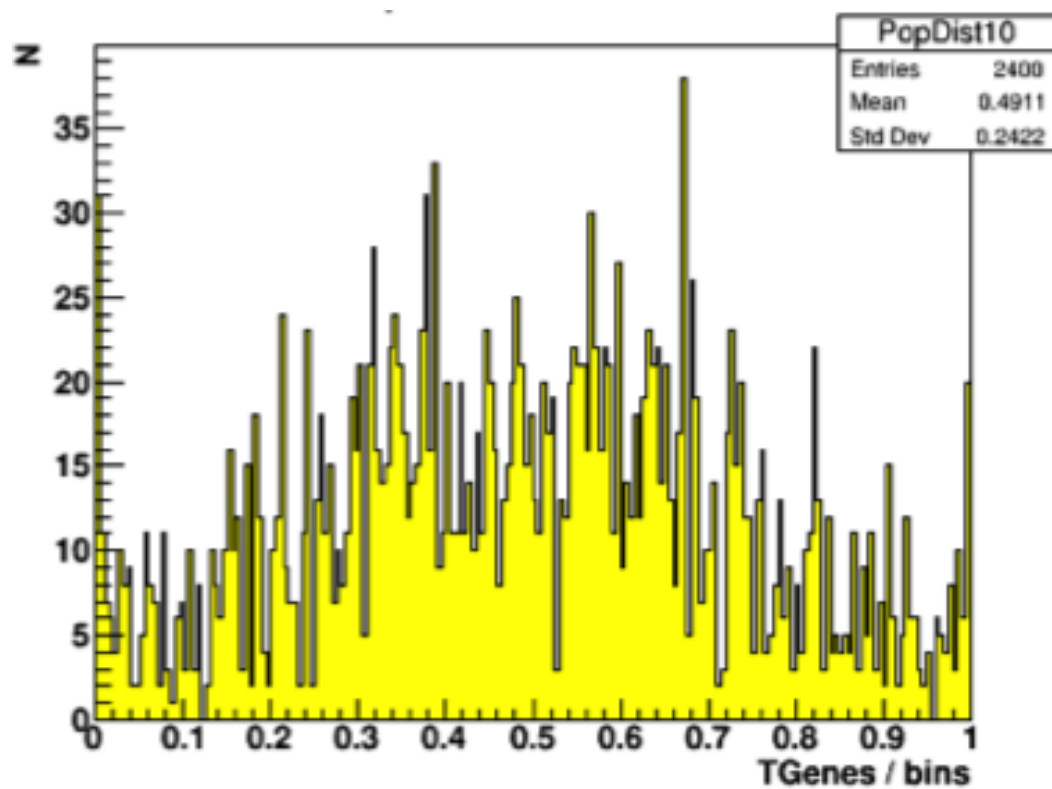


Рисунок 3.14 – Розподіл популяції для 10-го покоління для тесту DTLZ4

(алгоритм NSGA-II)

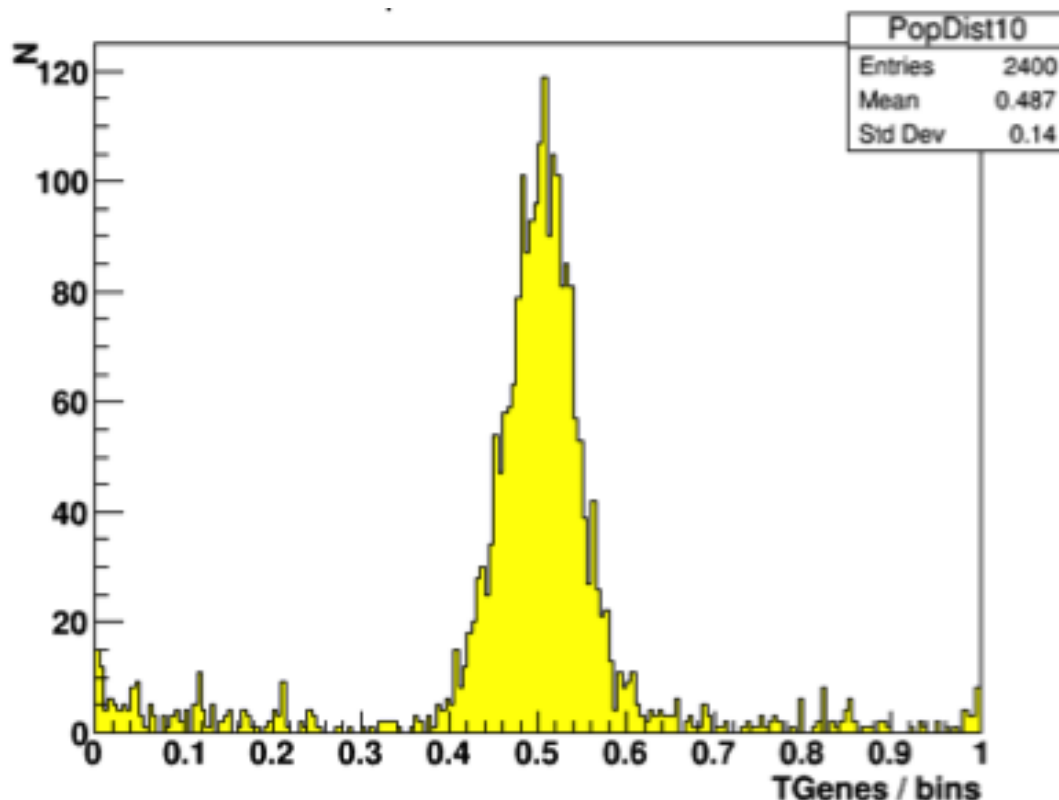


Рисунок 3.15 – Розподіл популяції для 10-го покоління для тесту DTLZ2 (для алгоритму NSGA-II - UPCA2)

На рисунку 3.10 та рисунку 3.11 представлені розподіли параметрів (середнє та стандартне значення відхилення (дисперсію)) та поведінку значення вартості залежно від використовуваних алгоритмів.

Порівнюючи рисунки 3.10, 3.12 і 3.14, в яких приведені результати застосування НГК-оператора, з рисунками 3.11, 3.13 та 3.15, де приведені результати без застосування НГК-оператор, можна спостерігати швидку збіжність до ідеальних значень параметрів задачі у першому випадку. Рисунки 3.11, 3.13 показують наближення до Парето-фронту у поєднанні з правильним набором параметрів. Середнє та стандартне значення відхилення (дисперсія) було значно зменшено та розглядаючи поведінку значення

вартості залежно від використовуваних алгоритмів можливо помітити зміну форми розподілу, що наближається до нормального розподілу (розподіл Гауса). Більше результатів для порівняння включення НГК-оператора за методом NSGA-II - UPCA2 і без включення при застосуванні для тестів DTLZ показано у додатку Б: рисунки Б.1 – Б.19.

Для нового алгоритму протестованого разом з новим оператором, ми представляємо перші попередні результати тестів DTLZ2, DTLZ4.

Рисунок 3.10 показує наближення до Парето-фронту та розподіл популяції 10-го покоління для алгоритму NSGA-II та для тесту DTLZ2 у поєднанні з правильним набором параметрів. Рисунок 3.11 показує наближення до Парето-фронту та розподіл популяції для 10-го покоління для алгоритму NSGA-II - UPCA та для тесту DTLZ2 у поєднанні з правильним набором параметрів. Середнє та стандартне значення відхилення (дисперсію) було значно зменшено, що ще раз доказує ефективність нашого методу.

Рисунок 3.12 показує наближення до фронту Парето та розподіл популяції для 10-го покоління для алгоритму NSGA-II та для тесту DTLZ4 у поєднанні з правильним набором параметрів. Рисунок 3.13 показує наближення до фронту Парето та розподіл популяції для 10-го покоління для алгоритму NSGA-II - UPCA та для тесту DTLZ4 у поєднанні з правильним набором параметрів. Середнє та стандартне значення відхилення (дисперсію) було значно зменшено, що ще раз доказує ефективність нашого методу.

Рисунок 3.10 показує розподіл популяції для 10-го покоління для алгоритму NSGA-II та для тесту DTLZ2. Рисунок 3.15 розподіл популяції для 10-го покоління для алгоритму NSGA-II – UPCA2 та для тесту DTLZ2 у поєднанні з правильним набором параметрів. Розглядаючи поведінку розподілення значення вартості популяцій залежно від використовуваних алгоритмів можливо помітити зміну форми розподілу, повторюючи більш нормальний розподіл (розподіл Гауса).

Висновки до розділу 3

У цьому розділі були приведені результати тестування нового НГК-оператора при застосуванні його в стандартних тестових багатоцільових задачах для генетичних алгоритмів. Проведено шляхом моделювання порівняльний аналіз використання різних варіантів введення методу головних компонент в генетичний алгоритм NSGA-II та за допомогою оцінки середньоквадратичної похибки знайдено оптимальний спосіб включення НГК-оператора (показаний на рисунку 3.7) який виявився найефективнішим для конвергенції алгоритму до Парето-фронту.

Результати досліджень за третім розділом опубліковані в статтях:

1. Шадура, О. Модифікація генетичних алгоритмів на основі методу нецетрованих головних компонент та стандартні тести. World Science, № 4(44)., стр. 4–11, DOI: 10.31435/rsglobal_ws (2019).
2. Shadura, O., Carminati, F. and Petrenko, A. Performance Optimization of Physics Simulations Through Genetic Algorithms, Journal of Computer Science, v.15, n.1, p. 57-66, DOI: 10.3844/jcssp.2019.57.66 (2019).

РОЗДІЛ 4

ТЕСТУВАННЯ ЗАСТОСУВАННЯ НГК-ОПЕРАТОРА В ОПТИМІЗАЦІЇ ПРОДУКТИВНОСТІ ПАКЕТУ МОДЕЛЮВАННЯ ТРАНСПОРТУ ЧАСТИНОК GEANTV

4.1 Програмний пакет Geant4

В цьому параграфі ми розглянемо сильні та слабкі сторони Geant4, як попередника програмного пакета GeanV [13], на прикладі його застосування для моделювання експериментальних подій в детекторі експерименту SHiP в ЦЕРНі, що було проведено дисертантом в команді співробітників колаборації SHiP [6].

Для підвищення обчислювальної продуктивності обробки експерименту важливо проаналізувати продуктивність Geant4 протягом усього циклу його розробки та використання, щоб забезпечити оптимальне використання обчислювальних ресурсів, а також сприяти поширенню передових методів для створення програмного забезпечення.

Для майбутніх експериментів на детекторі CMS на БАК передбачається, що обчислювальні потреби CMS збільшаться в 10 - 100 разів для модернізованого БАК з високою світимістю (HL-LHC) [19]. Конкретний фактор збільшення обчислювальних потреб буде залежати від розроблених підходів, зокрема ефективних методів оптимізації програмних пакетів обробки даних, вирішення майбутніх завдань моделювання, накопичення та реконструкції експериментальних подій.

Оновлення Geant4 у 2013 році, з можливістю багатопотокової обробки

подій, принесло значну економію пам'яті для обробки даних в експерименті CMS. Так, для програм CMS-моделювання, використання пам'яті було зменшено, наприклад, для першої події, 170 Мб і лише 30 Мб для обробки кожного додаткової події для кожного додаткового потоку. Але попри це, з введенням багатопотоковості загальна обчислювальна продуктивність роботи системи автоматизованого проектування і розрахунку Geant4 не поліпшилася та відхилення від ідеального масштабування складає приблизно на 10% гірше навіть при виконанні на 30-ти ядерному сервері.

Через оптимізацію коду та вдосконалення програмної платформи Geant4 і алгоритмів обробки подій, середнє підвищення продуктивності Geant4 протягом проведення експериментів на ВАК (2010-2015 рр.) було порядку 35% (рисунок 2.5).

Завдяки зусиллям команди Geant4, яка прагне вдосконалити обчислювальну продуктивність системи автоматизованого проектування і розрахунку, код Geant4 вже має високий рівень оптимізації, що ускладнює отримання додаткового підвищення продуктивності у майбутньому.

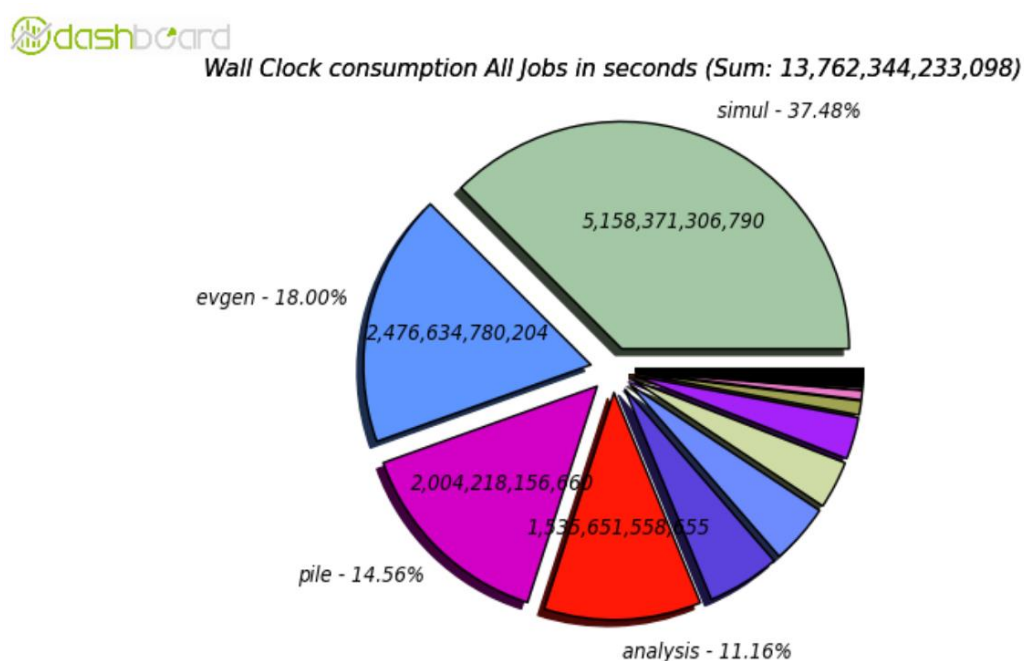


Рисунок 4.1 – Статистика з інформаційної панелі про відсоток часу

моделювання Geant4 на ресурсах WLCG

Ця ситуація була головною мотивацією для проведення досліджень в рамках цієї дисертаційної роботи, оскільки ми намагалися знайти нові інноваційні алгоритми для підвищення ефективності поточного стану симуляцій подій в експериментах у ФВЕ за допомогою програмного пакета GeantV [105]. Для дослідження сильних та слабких сторін Geant4 при оптимізації моделювання експериментів у ФВЕ дисертантом (в команді дослідників з колаборації SHiP) було проведене дослідження оптимізації процесів в детекторі експерименту SHiP, результати якого викладені в наступному підрозділі.

4.1.1 Geant4 та експеримент SHiP

Суперпротонний синхротронний прискорювач (SPS) є другою за величиною машиною в комплексі прискорювачів ЦЕРН. В нього попадають частинки з протонного синхротрона і потім прискорюються для створення пучків що використовуються у ВАК та в експериментах з фіксованою мішенню (експерименти NA61/SHINE [3] і NA62 [48]).

Експеримент SHiP («Пошук прихованих частинок») є новим експериментом в ЦЕРНі, що використовує фіксовану мішень на прискорювачі SPS. Він має на меті дослідити область енергій, де можна буде знайти сигнали від так званих прихованих частинок, які до цього часу не були знайдені на прискорювачах, наприклад, частинок темної матерії. Очікується, що елементарні приховані частинки, що передбачаються різними теоріями, можуть бути знайдені через розпад важких адронів.

Основною метою дослідження ФВЕ в експериментах SHiP є перевірка теорій з прихованими частинками та розширень Стандартної моделі, які включають довго живучі й дуже слабо взаємодіючі частинки (приховані

частинки), через пряме вимірювання їх розпаду в частинки зі Стандартної моделі. Експерименти проводяться на основі дослідження можливостей народження та реєстрації детекторами мезонів, які мають квантові числа чарівності та краси, але при цьому необхідно забезпечити рівень фону, що близький до нуля.

В цих експериментах оптимізація активного екрана мюонів для експерименту SHiP [6] сильно залежить від реалістичного опису багаторазового розсіювання мюонів в детекторі SHiP, яке описується в моделях програмного пакета Geant4.

4.1.2 Оптимізація мюонного щита для SHiP

Програмне забезпечення FairSHiP [53] для моделювання та реконструкції подій в експериментах SHiP базується на пакеті FairRoot [12], який повністю спирається на програмний пакет ROOT [28] і забезпечує базові C++ класи для побудови детектора використовуючи пакет ROOT TGeo [28] та використовується для аналізу подій. Для дослідження продуктивності фізичних симуляцій важливо зазначити, що фізичні події виробляються наступними генераторами Монте-Карло, а саме: PYTHIA8 [109] використовується для моделювання зіткнень протонів з фіксованою мішенню в експериментах SHiP, PYTHIA6 [108] для опису непружних взаємодій мюонів, та для моделювання реакції в детекторі SHiP використовується Geant4 [5].

На рисунку 4.2 схематично показано конфігурацію поля магнітних щитів для експерименту SHiP. Перша ділянка мюонного щита починається в частині мішені, що поглинає адрони та містить котушку, яка намагнічує залізний захисний блок, і потім продовжується до ділянки де розташовані магніти довжиною близько 20 м.

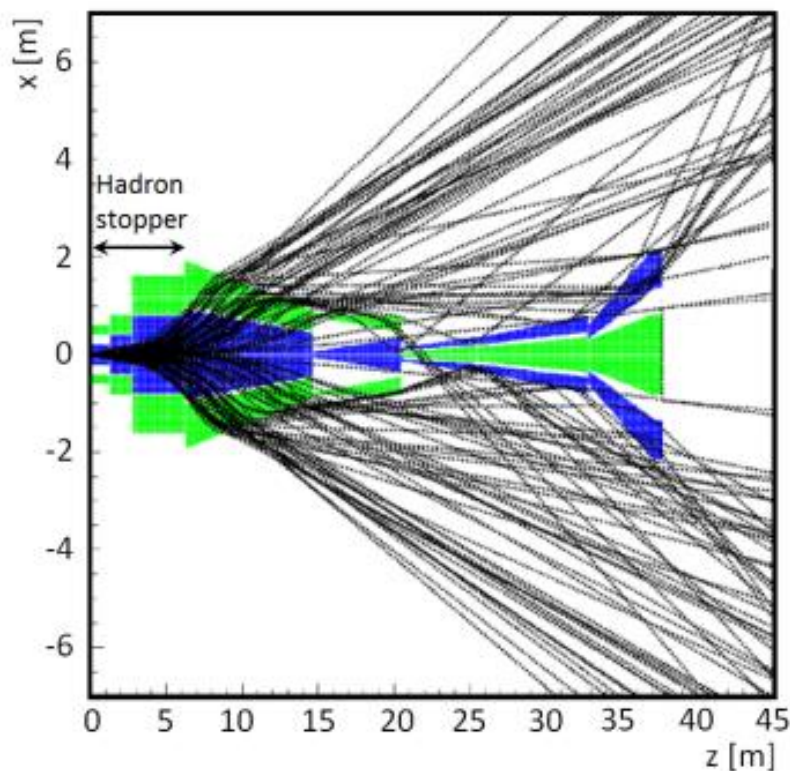


Рисунок 4.2 – Конфігурація поля магнітних щитів експерименту SHiP

Метою дослідження в статті [6] було проведення оптимізації процесів в детекторі SHiP, використовуючи повне моделювання експериментальних подій з застосуванням програмного пакета FairSHiP [53], для того щоб оптимізувати продуктивність з урахуванням її вартості та забезпечити стійкість процесів детектування шляхом оптимізації для менших значень напруженості магнітного поля.

При цьому необхідно було забезпечити контроль за наступними особливостями задачі оптимізації:

- може не вистачати достатньої кількості симуляцій для повної статистики;
- може не вистачати обчислювальної потужності для використання повного моделювання для оптимізації;
- майже дві однакові конфігурації можуть мати дуже різну

продуктивність;

- при оптимізації оцінка експериментальних точок є дуже місткою за ресурсами, інформація про градієнти недоступна і не може бути апроксимована;

- навіть при простій параметризації ми маємо ~ 50 вільних параметрів (довжини), кожен з яких змінюється від сантиметра до метра.

Фактична конфігурація магнітного поля для всього мюонного щита була оптимізована за допомогою методів машинного навчання з використанням великої вибірки мюонів з повним моделюванням (за допомогою Geant4 [5] зіткнення $2 \cdot 10^{10}$ протонів з мішенню SHiP. Проводилися інженерні дослідження з вивчення оптимальних методів складання конфігурації щита. Загальна маса мюонних щитових магнітів становила близько 1500 тонн. Поточна конструкція щита дозволяє знизити швидкість залишкових мюонів (реконструйованих у спектрометрі SHiP) зі швидкості вище $1\text{GeV} \cdot c$ до прийнятної величини.

В результаті цих досліджень, вдалося значно знизити вагу і вартість SHiP-детектора. В якості додаткового результату було отримане оптимальне значення продуктивності детектора (див. рисунок 4.3) зі значно зменшеною величиною магнітного поля, що доводить ідею про можливість оптимізації даних за допомогою додатків симуляції транспорту подій. Ідея викотана в [6], показала нові можливості роботи Geant4 за допомогою методів машинного навчання.

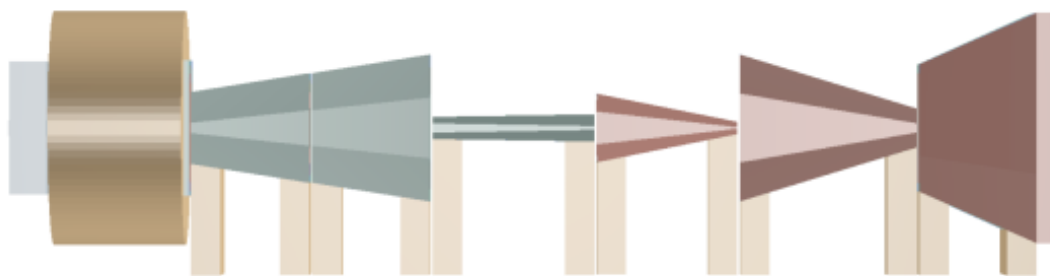


Рисунок 4.3 – Оптимальне значення продуктивності детектора

4.1.3 Оцінювання рівня сигналу та фону в експерименті SHiP

Для оцінювання ефективності роботи детектора в експерименті SHiP та виявлення очікуваного сигналу від прихованих частинок над фоном [6] проводилися симуляції роботи детектора шляхом моделювання його роботи в пакеті FairShip [53]. Народження чарівних адронів, що виникали від зіткнення протонів (з імпульсами 400 GeV/c) з фіксованою мішенню, моделювалося за допомогою Pythia6 [108], а симуляція розповсюдження частинок в матеріалах для різних конфігурацій щита та для різних фіксованих мішеней – за допомогою системи автоматизованого проектування і розрахунку Geant4 [5].

Розглянемо більш детально методологію проведення симуляцій Geant4. Для цього при моделюванні експерименту SHiP було застосовано метод *FixedTargetGenerator*. Він служить трьома основними цілями: а) забезпечує мюони, що залишають поглинач адронів, для подальшої обробки на різних установках активного щита мюонів; б) забезпечує нейтрино (електрон/мюон/тау) для досліджень фону в експерименті SHiP і для реконструкції сигналу в тау-нейтриному детекторі; в) забезпечує будь-які інші частинки, що залишають поглинач адронів, для вивчення потенційних фонових подій. Цей метод інтегрований у програмне забезпечення FairSHiP, що засноване на пакеті FairRoot [12].

Події зіткнення протонів з фіксованою мішенню можна або моделювати, використовуючи безпосередньо Geant4 [5], стріляючи протоном з імпульсом 400 GeV/c в мішень SHiP, або моделюючи події розсіяння першого протону на протоні або протона на нейтроні в фіксованій мішені, використовуючи генератор каскадних подій Pythia8 [109]. Потім народжені частинки транспортувалися в матеріалі детектора за допомогою Geant4, де проводилося зчитування з файлу готових для використання даних, згенерованих для чарівних або красивих адронів, за допомогою генератора каскадних подій Pythia6 [108].

Частинки можуть транспортуватися в пакеті Geant4, якщо їхня кінетична енергія вище певного порогу. Це обмеження необхідне для того, щоб прискорити моделювання і генерувати більш енергетичний фон за менший час роботи процесора. Всі частинки, які залишають адронний поглинач, записуються в файл, якщо їхня кінетична енергія перевищує поріг.

Додатковий варіант існує для зберігання тільки інформації про мюони. У всіх випадках кінематика нейтрино та P/P_T кореляції зберігаються в 2-мірних гістограмах. Також зберігаються історії всіх збережених частинок, зокрема материнських частинок та коду процесу в Geant4, який виробляв частинку.

Оцінка площини вирівнювання здійснюється C++ класом *exitHadron-Absorber*, що знаходиться у пакеті FairSHiP у директорії *muonShieldOptimization*. Площина розташована або після «MuonShieldArea1/MagnAbsorb 2MagCRB1» (див. рисунок 4.5), який є одним з магнітів всередині адронного поглинача експериментальної установки SHiP, або в *fzPos* параметрах, який повинен бути наданий через *SetZposition()* метод.

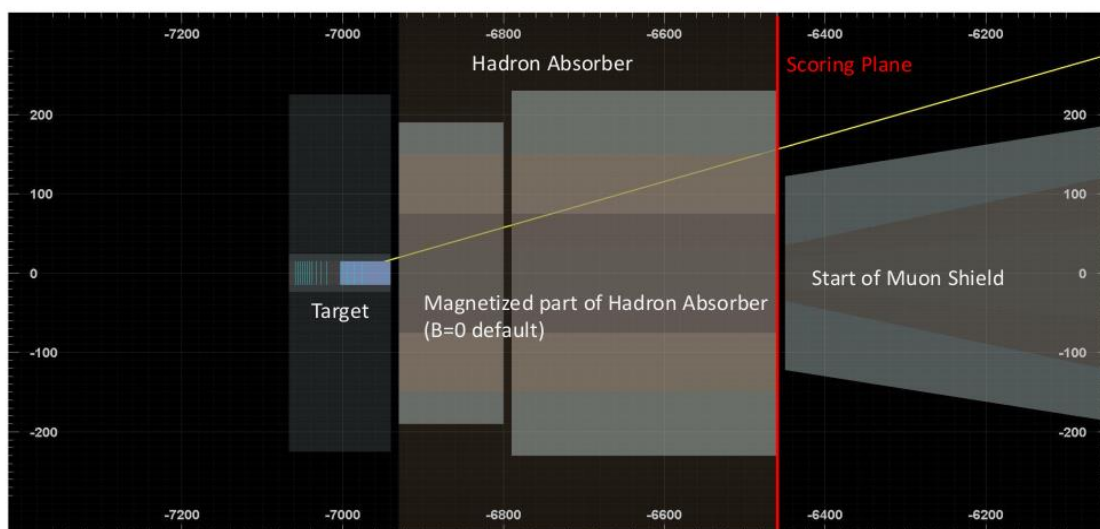


Рисунок 4.5 – Адронний поглинач для експериментальної установки SHiP

Клас *exitHadronAbsorber* взаємодіє з Geant4 двома способами. По-перше, Geant4 буде викликати для будь-якої частки знову створений метод *exitHadronAbsorber::PreTrack*. Якщо кінетична енергія нижче певного порогу, *SetEnergyCut()*, транспорт цієї частинки припиняється. По-друге, для будь-якої частинки, що досягає площини скорингу, викликається метод *AddHit()*. Він записує, положення і імпульс, коли частинка потрапляє в площину, а також всю інформацію про частинку, коли вона була створена.

Якщо кінетична енергія частинки знаходиться вище порогового значення, інформація зберігається як об'єкт *vetoPoint* у контейнері *vetoPoint*, разом з повною історією частинки, її матері, бабусі і т.д., і як об'єкти *MCTrack* в контейнері *MCTrack*. Існує додаткова опція, *SetOnlyMuons*, для зберігання інформації тільки для мюонів. Крім того, інформація про імпульс нейтрино і P/P_T кореляції зберігаються в 2-мірних гістограмах для подальшої обробки пакетами *Genie* і *GenieEventGenerator* [18].

C++ клас *FixedTargetGenerator* надає вхід для Geant4, додаючи частинки до стека з інформацією про ідентифікатор частинки, імпульс, положення, час польоту, індекс матері частинки у стеку, вагу та ідентифікатор процесу, який створив частинку. Якщо кінетична енергія частинки перевищує заданий поріг,

встановлюється *SetEnergyCut()*, а Pythia8 позначає її як кінцеву частинку, так вона позначається для подальшої обробки Geant4.

Після того, як *FixedTargetGenerator* завершив обробку заданого числа подій, починається етап постобробки для очищення результату від порожніх подій, видаляючи всі події без частинок, що досягають площини вирівнювання. За цими результатами створюється ROOT файл [28], з назвою яка відповідає кількості моделювань зіткнень протона з мішенню (ЗПМ) для цього файлу. Збережене ROOT дерево [28] містить два контейнери, один *cbmsim/vetoPoint* з записами даних з площини вирівнювання, і *cbmsim/MCTrack* з даними частинки, що потрапляє в площину вирівнювання і з даними всіх інших частинок, що пов'язані з нею. Це ROOT дерево [28] може бути використано для подальшої обробки мультиплексним генератором *muonsBackgroundGenerator*. Збережені гістограми кінематики нейтрино можуть бути використані в якості вхідних даних для генерування нейтринних непружних подій за допомогою генератора Genie [18].

Розглядаючи типову SHiP симуляцію було отримані перші результати оцінки ефективності Geant4 [6]. Вони показали непогані результати в порівнянні з Pythia6/Pythia8, але дозволили дослідити негативні сторони Geant4. Відмітимо, що вадами програмного пакета Geant4 є те, що він витрачає велику кількість процесорного часу для розрахунків у ФВЕ, був розроблений переважно більше ніж 10 років тому та не демонструє жодної гнучкості в оптимізації часу обчислення для задач моделювання.

Проведене дослідження оптимізації процесів в детекторі експерименту SHiP, допомогло оцінити сильні та слабкі сторони Geant4 при оптимізації моделювання експериментів у ФВЕ (в команді дослідників з колаборації SHiP) та оцінити дизайн налаштування Geant4. Одним з найбільших недоліків є його монолітність як програмного додатка, що унеможливорює будь-яку оптимізацію та відсутність параметризації методів класів у Geant4. На протязі останнього десятиліття розвиток Geant4 [5] був направлений лише на його

застосування для традиційного повного моделювання детекторів в експериментах на БАК. Аналіз типових симуляцій в експерименті SHiP приводить також до висновку, що слабкими сторонами Geant4 є складний дизайн, що містить глибокі ієрархії різних структур даних та складна система функціональних викликів, які важко експлуатувати на найкращих сучасних архітектурах, через низькі значення показників ефективності інструкцій та не ефективне використання цим додатком кеш-пам'яті.

4.2 GeantV – наступне покоління програмних пакетів для моделювання транспорту частинок

Проект GeantV [13] вважається програмним забезпеченням наступного покоління для симуляцій транспорту частинок при обробці даних експериментів з ФВЕ на Великому адронному колайдері (ЦЕРН, Женева).

GeantV [13], у порівнянні з Geant4, має більш гнучку структуру, що призводить до кращих можливостей широкої підтримки ним різних обчислювальних архітектур і кращої масштабованості для складних обчислень.

Однією з головних особливостей GeantV [14] є застосування метода векторизації для обробки даних експериментів. Ідея використання векторизації стає дуже популярною в ФВЕ. Невекторизований код використовує близько 25% потужності сьогоденішнього обладнання. Він виконує інструкції по кожному набору даних для вхідних даних, замість виконання спеціалізованих SIMD (англ. single instruction, multiple data – одиночний потік команд, множинний потік даних) інструкцій на векторі набору даних (для тих же вхідних даних), але у тій же кількості циклів. Оскільки векторизований код загалом більш компактний (через операції над векторами), він зазвичай також покращує продуктивність кешу у кеш-пам'яті.

Векторизація підвищує ефективність (і зокрема пропускну спроможність) програмного забезпечення ФВЕ, використовуючи векторні реєстри центрального процесора. Особливо це важливо для додатків моделювання роботи детектора, які є одним з основних споживачів обчислюваних ресурсів у ЦЕРН [88].

Відмітимо, що покращення продуктивності при векторизації можливо дуже легко помітити, коли всі дані зберігаються в кеш-пам'яті. Але якщо дані потрібно часто переміщати з пам'яті в кеш-пам'ять, то підвищення продуктивності стає непомітним, як це можна побачити у таких випадках, як обробка великого розміру масиву даних. Тут витрати, які пов'язані з переміщенням даних, домінують настільки, що ефект від векторизації практично не помітний, або векторизація може мати навіть несприятливий ефект.

Нові можливості функціональності ядра GeantV [15] полягають в тому, щоб забезпечити створення оптимальних структур даних та розпаралелювання роботи та керування робочими потоками. Це можна реалізувати, забезпечивши когерентність роботи системи автоматизованого проектування і розрахунку шляхом пакування параметрів частинок у вектори та використання цих векторів для моделювання за SIMD-оптимізованими алгоритмами, враховуючи специфіку моделювання геометрії детектора та фізики транспорту частинок у детекторі.

Для запуску симуляції в пакеті GeantV [15, 14, 13] користувач налаштовує конфігурацію, використовуючи різні набори параметрів, такі як розмір буфера подій, розмір векторів або різні варіанти паралельного опрацювання даних в GeantV. GeantV надає можливість керувати фіксованою кількістю спільних робочих потоків за допомогою «пропагаторів-поширювачів» з метою збільшення кількості оброблених даних та забезпечені локальності даних при одночасному зменшенні конфліктів в цьому процесі. Локальність коду – дуже важливий аспект коду, що відповідає

стратегії ефективного керування кешем у кеш-пам'яті та стосується принципу локальності, метою якого є розміщення пов'язаних даних в пам'яті для забезпечення ефективного кешування для організації пам'яті NUMA (англ., Non-Uniform Memory Access – архітектура з неоднорідним доступом до пам'яті). GeantV може використовувати, як і окремі потоки та завдання, так і кластери потоків на одному і тому самому процесорі використовуючи організація пам'яті NUMA та використовувати знання топології обладнання для оптимального конфігурування розподілу ресурсів [27].

4.2.1 Визначення багатокритеріальної задачі для моделювання з використанням CAE-системи GeantV

Моделювання за допомогою CAE-системи GeantV являє собою складну систему взаємодії різних програмних модулів, необхідних для забезпечення симуляції транспорту частинок, яка включає транспортування (евристичний алгоритм транспортування частинок) з використанням різного набору навігаторів, політик планування, фізичних моделей з таблицями поперечних перерізів взаємодій частинок в різних реакціях, а також модельні бібліотеки з геометричними алгоритмами [21].

Для ефективної роботи великих програмних додатків, таких як GeantV, що налаштовані на використання обчислювальних стратегій SIMD та паралельного моделювання, також необхідно враховувати складність процесорів та архітектур, що використовуються для моделювання.

З цих причин, моделювання завжди повинно бути оцінено в порівнянні до значення інтенсивності його експлуатації, та в оцінці якості програмного продукту з точки зору моделювання транспорту частинок. Це породжує питання пошуку балансу продуктивності для операцій для чисел з рухомою комою, використання оптимальної пропускної здатності для пам'яті,

специфікації ієрархії кешування, оцінка прискорення моделювання при зміні рівня паралелізму, який залежить від архітектури, що використовується для виконання завдань симуляції, а також оцінки всіх етапів моделювання у питанні прийнятності для користувача рівня витрат часу і споживання пам'яті для обробки завдань реконструкції подій в ФВЕ.

Ці завдання класифікуються, як задача оптимізації «чорного ящика» за допомогою пошуку екстремального набору точок, та точкових оцінок для значення функцій.

Завданням оптимізації для «чорного ящика» є мінімізувати векторну функцію $f(\vec{x}): \mathbf{R}^n \rightarrow \mathbf{R}$, для якої аналітична форма невідома.

Оптимізацію «чорного ящика» можна провести на двох рівнях, отримавши:

- оцінку значень функції, без похідних і градієнтів (коли градієнтні методи не доступні);
- приблизні градієнти функції (що взагалі потребує великої кількості оцінок):

$$\frac{d f(x)}{d x} : \Delta_k f \approx \frac{f(x_k + \varepsilon_k) - f(x_k)}{\varepsilon_k} \Delta x_k.$$



Рисунок 4.6 – Проблема чорного ящика

У дисертації ми розглядаємо симуляцію GeantV [16] як задачу «чорного ящика» з набором фітнес функцій. Для оптимізації симуляції GeantV, нам потрібно узагальнити набір цілей, які ми хотіли б досягти за допомогою процедури оптимізації та врахувати їхню залежність від великої

кількості векторних параметрів:

- оптимізація загального часу виконання прикладів симуляції та середнього часу для створення однієї популяції;
- зменшення використання пам'яті в додатках;
- оптимізація кількості викликів інструкцій для всього моделювання, специфічного для поточної архітектури;
- забезпечення цілісності даних та своєчасної синхронізації змін в даних під час роботи симуляції (оптимізація NUMA і т. д.);
- максимізація операційної інтенсивності моделювання.

4.2.2 Огляд прототипу GeantV

Поведінка GeantV [16] повинна бути попередньо налаштована перед етапом ініціалізації. Набір загальних функцій має бути налаштований під час компіляції за допомогою команди ***cmake – D[option]***, де доступні наступні опції: опція ***USE_ROOT*** включає підтримку ROOT [28] (за замовчуванням ON); опція ***WITH_GEANT4*** – включає приклади Geant4 (за замовчуванням ON); опція ***USE_NUMA*** дозволяє виявлення топології і використовувати технологію NUMA для архітектур з неоднорідним доступом до пам'яті, яка характеризується організацією пам'яті в паралельних архітектурах EOM та в якій над фізично розподіленою між окремими обчислювальними блоками пам'яттю створюється спільний адресний простір (за замовчуванням ON), вимагає використання пакету hwloc > 1.11 [27] та опція ***USE_TBB*** – включає приклад CMSToyGV (за замовчуванням OFF), вимагає Intel Threading Building Blocks (також відома як TBB, кросплатформна бібліотека шаблонів C++, розроблена компанією Intel для паралельного програмування).

Під час запуску, GeantV може бути налаштований з використанням структури GeantConfig, що дозволяє описати кілька параметрів, таких як:

кількість подій, які необхідно промодельовувати, максимальну кількість подій в буфері, що можуть бути оброблені одночасно, режим роботи (автономний проти зовнішнього керування) та розмір «кошика» подій (контейнер, що використовується при транспортуванні частинок). Крім того, для геометричних алгоритмів та для конфігурації магнітного поля для фізичних моделей може бути включена специфічна для GeantV функція, яка називається «процедура пакетування» з використання обчислювальних стратегій SIMD/MIMD [16].

CAE-система GeantV використовує концепцію векторизованої обробки для транспорту частинок на основі використання процедури «пакетування», яка створює вектори частинок для конкретної геометрії з числами з рухомою комою та з урахуванням фізики явищ. Мета полягає в тому, щоб використовувати векторні реєстри, доступні у сучасних процесорах, і оптимізувати використання кешу у кеш-пам'яті, і водночас максимально використовувати геометричні та фізичні алгоритми.

Процедура пакетування є одним з основних компонентів CAE-системи GeantV, що дозволяє попередньо вибирати і об'єднувати треки, з якими будуть проводити одні й ті ж кроки обробки, щоб поліпшити кешування і дозволити багаточастинкову векторизацію. Зокрема:

- моделювання організується поетапно, наприклад: обчислення довжини обчислювальної взаємодії частинок, запити для геометричних та фізичних алгоритмів, інші «покрокові» дії, що викликають користувальницькі програми;

- кожен етап може бути розділений на декілька категорій обробки: у випадку геометрії – це треки, що знаходяться в різних логічних томах (одиниця геометрії детектору), обробляються окремо; у випадку стадії розповсюдження – це нейтральні треки що обробляються інакше, ніж заряджені;

- кожна інша категорія обробки на етапі має відповідну одиницю

виконання роботи, «worker або робітник», що реалізований як пара різних методів DoIt(), що мають різні скалярні та векторні сигнатури;

- кожному окремому «робітнику» можна додатково пройти процедуру пакетизації, використовуючи паралельну службу під назвою «пакетувальник», яка буде приймати в якості вхідних даних: показники треків і заповнювати їх, як тільки «кошик» подій буде заповнений до налаштованого конфігурацією розміру (який розглядається як параметр оптимізації);

- пакетування може бути включено для кожного «робітника» для обробки даних і який може векторизувати масиви треків для частинок, причому у поточній версії «кошик» подій можна зробити також для геометричних алгоритмів та методів для генерації магнітного поля.

CAE-система GeantV має на меті забезпечити довгострокове рішення для задач моделювання транспорту частинок, що працюють в загальноприйнятих процесорах, прискорювачах, таких як графічні процесори та Xeon Phi 7120X Knights Landing, а також в середовищі для високопродуктивних обчислень [13].

Хоча концепція прототипу GeantV в основному була розроблена та апробована для загальноприйнятих процесорів, було також передбачено використання прискорювачів в GeantV, що реалізовано у вигляді архітектури або технології для кожного конкретного бекенда [13]. Цей підхід дозволяє абстрагувати не тільки основні об'єкти, такі як скаляр або вектор, але також формалізувати загальні програмні ядра (англ., kernel, – функції (завдання), що будуть виконуватися на GPU) для обчислень, використовуючи прозорі бібліотеки або конкретні структури даних для об'єктів, що засновані на використанні декількох провідних бібліотек в цій галузі.

Важливою особливістю CAE-системи GeantV є можливість надавати оптимізовані структури даних та паралельні служби для них, такі як треки що зберігаються в AOS/SOA форматі (англ., Array of Structures (AoS)/Structure of Arrays (SoA) – масив структур/структура масивів), управління чергами,

«пакетувальники» та контроль паралельних робочих потоків використаних для організації симуляції [14]. Для виконання своїх задач GeantV використовує наступні процедури:

- організація балансування роботи, перемикання між векторними і скалярними режимами, пріоритизація транспорту подій певного типу;
- забезпечення когерентності роботи шляхом «пакування» частинок у вектори;
- групування локальності коду за критеріями: геометрія, фізика (локальність коду – дуже важливий аспект коду, що відповідає стратегії ефективного керування кешем у кеш-пам'яті та стосується принципу локальності, метою якого є розміщення пов'язаних даних в пам'яті для забезпечення ефективного кешування);
- забезпечення інструментів векторизації в алгоритмах геометрії та фізики.

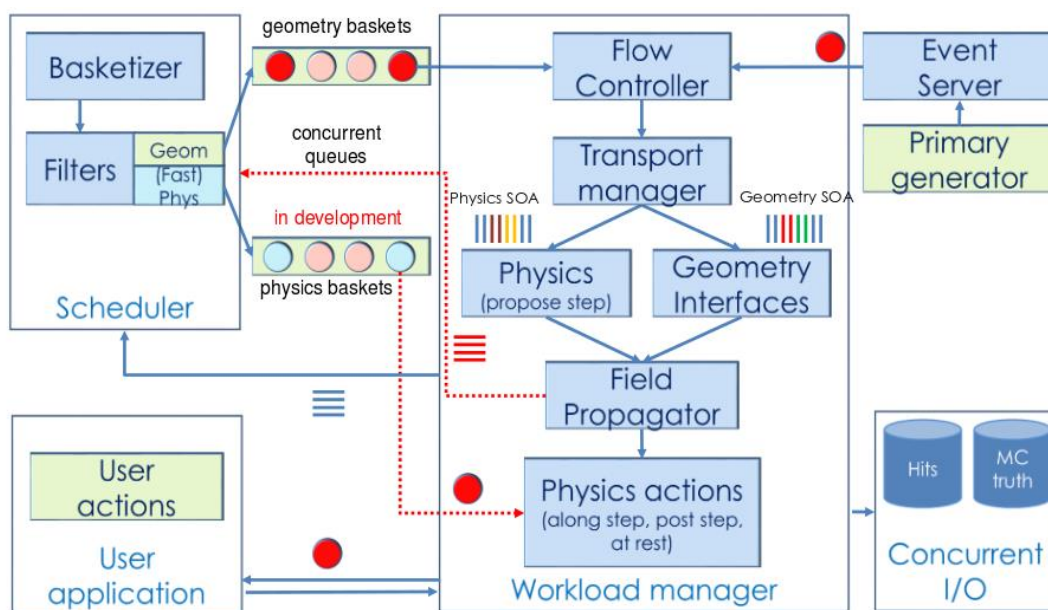


Рисунок 4.7 – Функціональна діаграма алгоритму GeantV

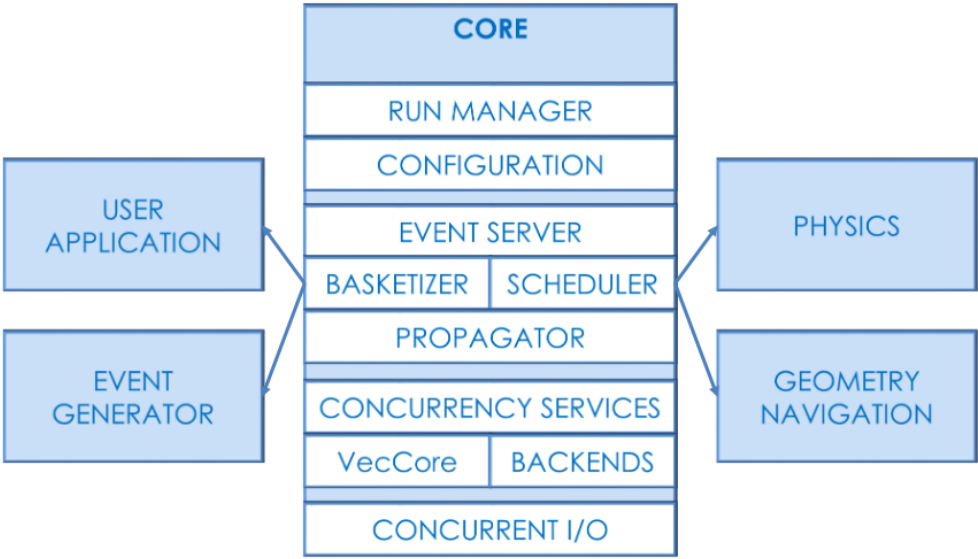


Рисунок 4.8 – Схема основного модуля GeantV

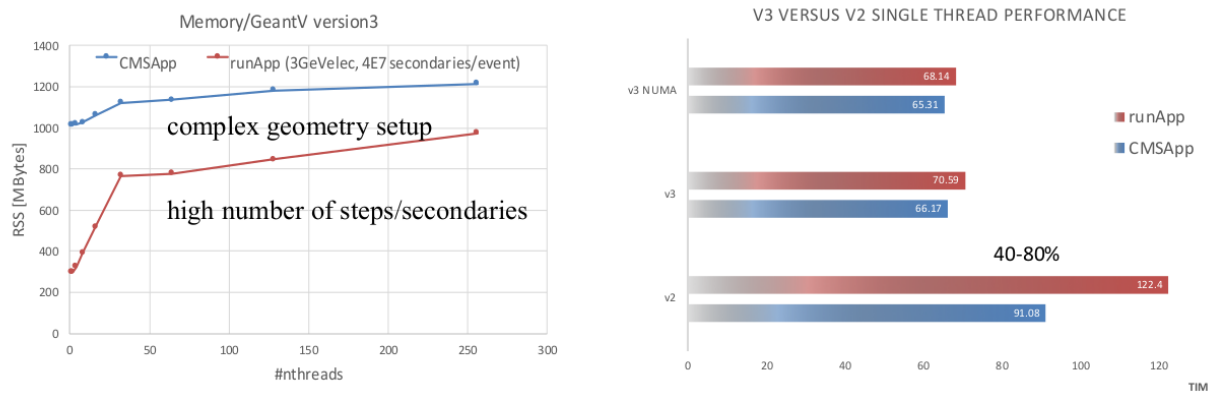


Рисунок 4.9 – Використання пам'яті (RSS) GeantV додатком у залежності від кількості потоків

можуть бути або згруповані з іншими треками та оброблені за допомогою векторизованих алгоритмів, або обробляються обраною моделлю в скалярному режимі. Такий підхід покращує локальність та узгодженість кешу у кеш-пам'яті в порівнянні з класичними підходами на основі стеку (багаторівневий підхід).

Основною функціональністю ядра програмного додатка GeantV [23] є:

- надання відповідних типів для структур даних та інших сервісів;
- створення та контроль за паралелізацією робочого процесу;
- забезпечення когерентності роботи шляхом «пакування» частинок у вектори частинок;
- забезпечення алгоритмів векторизації для геометричних та фізичних алгоритмів.

4.2.3 Модель паралелізму GeantV

Для проекту GeantV очікується, що покращена пропускна спроможність для подій, компенсує зростаючу потребу в моделюванні даних для режиму роботи з HL-LHC [19], який буде генерувати величезну кількість подій. Зазначимо, що GeantV має значні переваги через використання оптимізованих компонентів, таких як нова бібліотека моделювання геометрії VecGeom [21], яка забезпечує кілька нових функцій, включаючи векторизацію та прозорий доступ до новітніх комп'ютерних архітектур.

Бібліотека VecGeom [22, 21] здатна обробляти вектори треків частинок, що чудово адаптуються до багатопоточних завдань. Однією з переваг VecGeom в порівнянні з подібними бібліотеками є висока портативність для використання на прискорювачах (графічних процесорах та для Xeon Phi 7120X - Knights Landing та ін.).

Обговорюючи оптимізацію моделі GeantV, ми можемо відмітити вісім

рівнів паралелізму роботи GeantV [67], причому «дрібнозернистий», англ. “fine-grained” паралелізм є єдиним засобом, який дозволяє отримувати додаткові ефекти як у пропускній здатності, так і у часі для отримання розв’язок задачі оптимізації. Інші методи мають свої сильні та слабкі сторони: паралелізм на рівні апаратної системи дає дуже слабкий ефект, тоді як багатоядерний паралелізм дозволяє зменшити об’єм використаної пам’яті та час для розв’язку, але не вирішує проблему скорочення пропускної спроможності. З іншого боку високопродуктивні обчислення дають можливість запускати багатопаралельні обчислювальні завдання з високою пропускною спроможністю. Ці різні параметри паралелізму можуть бути включені в GeantV [20] і збільшують розмірність простору оптимізованих параметрів. Наша ідея полягає в тому, щоб спробувати оптимізувати моделювання, використовуючи спеціальний набір інструментів для машинного навчання, намагаючись оптимізувати збільшення обсягу завдань для обробки даних [104].

Основними характеристиками функціональності ядра системи автоматизованого проектування і розрахунку GeantV є надання оптимізованих структур даних та паралельних служб для керування паралельними робочими потоками, забезпечення когерентності роботи шляхом «пакування» частинок у вектори та використання цих векторів в алгоритмах геометрії та фізики.

Диспетчер запуску GeantV – це модуль, який реалізує задану конфігурацію запуску та параметрів моделювання. Диспетчер запуску також керує циклами та потоками оброблюваних подій, інжекцією згенерованих подій у систему, параметрами паралельності оброблення даних, зокрема такими, як (тимчасове) зупинення або завершення роботи.

Як передумова для запуску симуляції [20], користувачеві потрібно буде створити об’єкт конфігурації, який може бути налаштований за допомогою різних наборів параметрів, таких як розмір буфера подій, розмір вектора або декілька варіантів налаштування паралельної роботи GeantV. Наші

інструменти для налаштування симуляцій GeantV [14] – це конфігурації набору параметрів, що працюють на вузлах та входять в матрицю параметрів, що сформована на результатах спостереження за планувальником та характеристиками вузлів, і які можна оцінити за допомогою матриці функцій фітнесу (рисунок 4.8).

Подальше вдосконалення ефективного управління симуляціями можна було б запровадити з точки зору управління подіями, виходячи з параметричної настройки еволюційного алгоритму для моделювань та введення в дію сервера диспетчеризації подій (рисунок 4.10). Важливим для використання є застосування нової політики балансування подій у середовищі, що складається з декількох вузлів та може бути вирішена за допомогою стратегії, яка допомагає зменшити час під час виконання завдань на менш ефективних вузлах обробки.

4.2.4 Ключові елементи основної інфраструктури GeantV

Для запуску користувачу GeantV потрібно створити об'єкт конфігурації для налаштування GeantV: налаштувати параметри кількості подій, розмір буфера подій, розмір вектору, геометрію, фізику та параметри паралелізму: максимальну кількість дозволених процесів, кількість «пропагаторів-поширювачів» що відповідають за функцію розподілення та балансування обробкою подій, організувати модель паралелізму для симуляції.

Наступним кроком є створення користувацької програми та генератора фізичних подій. Також важлива роль полягає у створенні менеджера запуску та ініціації циклу подій GeantV.

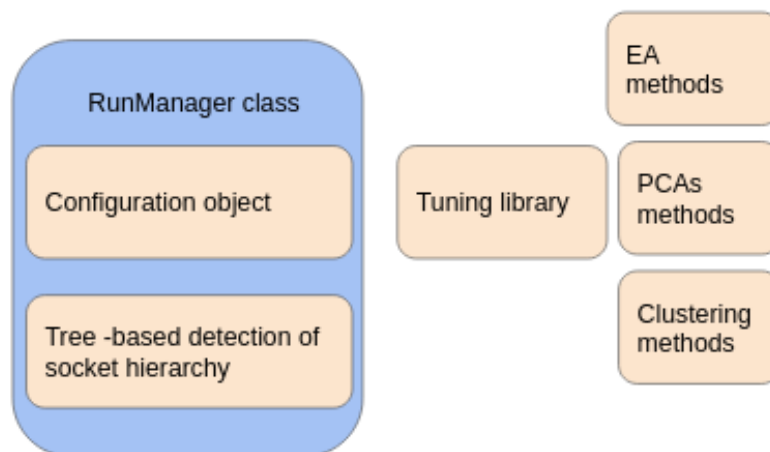


Рисунок 4.12 – GeantV Run Manager або менеджер запуску

Run Manager або менеджер запуску розглядається як точка входу для керування GeantV. Цей C++ клас був спеціально розроблений без можливості статичного доступу, а менеджер запуску доступний в GeantV через наявні інтерфейси. *Run Manager* або менеджер запуску також використовується для ініціалізації геометрії, управління циклом подій (внутрішнього або зовнішнього), ініціалізацією списку «пропагаторів-поширювачів» та завершенням роботи (рисунок 4.12). Цікавою особливістю є те, що GeantV може навантажити обчислюваний процесор за допомогою або одного або декількох пропагаторів-поширювачів. Метою цієї функції є збільшення кількості даних що будуть оброблені та покращення локальності коду. «Пропагатори-поширювачі» створюють свої власні черги з власними «кошиками» подій. Коли «пропагатори-поширювачі» стають вільними (тобто всі події оброблені), тоді їх черга обробляється менеджером запуску на предмет нових даних для обробки.

4.2.5 Налаштування параметрів для багатокритеріальної задачі оптимізації CAE-системи GeantV

В якості вхідного вектора або індивідууму для генетичного алгоритму, ми використовуємо наступний набір параметрів для GeantV:

а). Параметри обробки подій:

- загальна кількість подій;
- кількість подій в буфері;
- поріг для визначення пріоритетів подій;

б). Параметри паралельності:

- кількість потоків для використання паралельності процесора;
- політика планування для оптимізації локальності
- планування NUMA локальності для GeantV додатку.

в). Розмір вектору:

- вихідний розмір вектору;
- максимальний дозволений розмір вектора;

г). Специфічні параметри настройки:

- кількість кроків для алгоритму детекції важливих фігур для геометричних алгоритмів для детектора;
- порогове значення для перемикання між скалярним та векторним режимами роботи додатку;
- спеціальний поріг для активації диспетчеру завдань;
- порогове значення для контролю використаної пам'яті;
- порогове значення для знищення подій.

4.3 Високопродуктивні розрахунки у моделюванні за допомогою САЕ-системи GeantV та їх обчислювана складність

Прискорення розпаралелювання коду визначається як відношення («wallclock» часу для синхронізації)/(«wallclock» часу для паралельного виконання завдання). Це один з найпростіших і найбільш широко використовуваних показників для оцінювання паралельної програми. Паралельні «накладні» витрати – це час, необхідний для координації паралельних завдань, на відміну від здійснення корисної обчислюваної роботи. Паралельні накладні витрати можуть включати такі фактори, як:

- час запуску завдання;
- синхронізація;
- передача даних;
- накладні витрати, спричинені через використання паралельних мов, бібліотек, операційної системи тощо;
- час завершення завдання.

На жаль більшість моделювань є вже паралельними, але з поганим рівнем організації синхронізації, вирішуючи одночасно багато подібних, але незалежних завдань, що потребує координації між завданнями.

Аналізуючи обчислювальну техніку для високопродуктивних обчислень та процесів, ми також повинні згадати про інші важливі параметри, які можуть бути цікавими для оцінки якості робочого циклу, наприклад як масштабованість.

Масштабованість – це здатність паралельної системи (обладнання та/або програмного забезпечення) демонструвати пропорційне збільшення паралельного прискорення з додаванням більшої кількості ресурсів.

Фактори, що сприяють масштабованості, включають:

- сучасне апаратне забезпечення, особливо розмір параметрів

пропускної здатності пам'яті, а також процесор та властивості мережевого зв'язку;

- сучасні алгоритми застосування ресурсів;
- алгоритми зменшення накладних витрат.

GeantV використовує концепцію групування робочих вузлів. Кожною групою робочих вузлів керує C++ клас `geant::Propagator`. Групи під одним і тим же «пропагатором-поширювачем» співпрацюють для копіювання треків, але одночасно вони не взаємодіють з іншими робочими вузлами іншого пропагатора-поширювача.

Концептуальною одиницею роботи в GeantV є C++ клас `geant::Track`, що поширюється на всі активні потоки за допомогою синхронного обслуговування в `EventServer`. Якщо налаштована підтримка топології (NUMA), пам'ять для треків і CPU для робочих вузлів прив'язана до одного фізичного вузла на «пропагаторі-поширювачі». Користувач повинен визначити кількість «пропагаторів-поширювачів» і апаратних потоків для «пропагатора-поширювача» при створенні `RunManager`.

Зазвичай за правилом, розробники наполягають у створенні одного або більше «пропагаторів» на NUMA-вузол. GeantV може ефективно працювати також у багатопроцесорному режимі, створюючи процес на кожному обчислювальному вузлі, використовуючи один «пропагатор-поширювач» і стільки робочих вузлів, скільки потрібно для оптимального використання всього потенціалу фізичного вузла. Багатопроцесорний підхід має кращу масштабованість, але гірше використання пам'яті, ніж окремий процес.

Моделювання можна запустити в одному з двох режимів:

`geant::RunManager::RunSimulation` та `geant::RunManager::RunSimulationTask`.

У `geant::RunManager::RunSimulation` режимі, GeantV генерує фіксовану кількість робочих потоків, зберігаючи їх до кінця виконання. Події генеруються з використанням первинного генератора подій, а основні треки для подій будуть доступні через службу синхронного обслуговування

EventServer для всіх робочих вузлів. Треки можуть бути змішаними за допомогою буферизованих подій, і вони можуть бути альтернативно синхронізовані з іншого робочого вузла для одного і того ж «пропагатора-поширювача», якщо активовано режим баскетизації. Визначені користувачем параметри конфігурації, пов'язані з рівнем паралельності системи в цьому режимі, повинні бути узгодженими, кожен робочий вузол повинен мати деяке початкове навантаження. Кількість початкових треків, заданих сумою первинних треків для параметра кількості буферизованих подій, повинна перевищувати кількість запитаних у планувальника робочих вузлів.

У *geant::RunManager::RunSimulationTask* (зовнішньо керований цикл подій) режимі, зовнішня структура взаємодії з службою *geant::EventServer* у внутрішньому циклі подій, плануючи завдання для *RunManager::RunSimulationTask*. Вхідні події генеруються користувацьким фреймворком і зібрані в *geant::EventSet* структурі даних. Метод *RunSimulationTask* повторно запускається після заповнення набору подій, наданого в якості вхідних параметрів.

В дослідженнях GeantV [23] виконувалися тести для перевірки парадигм паралельного програмування (MPI) у гетерогенному кластері з загальним числом подій, розділених між всіма процесами. Попередні результати показали хороше масштабування, еквівалентне масштабуванню для одиничного вузла, що підтверджує можливість ефективної обробки завдань у багато серверному середовищі (у дата центрі).

4.4 Масивні паралельні обчислення і очікувані результати продуктивності

Концепція ґрид для ВАК, що дозволила запускати паралельно мільйони

завдань реконструкції подій в детекторах, базується на ідеї про те, що фізичні події загалом не корелюються. Попередні оцінки в ході нашого дослідження вказують, що ми можемо сподіватися на прискорення до 5 % при використанні сучасних процесорів для моделювання методом Монте-Карло для експериментів у ФВЕ. Це дозволить звільнити ресурси, що використовуються для задач з високопродуктивними обчисленнями, та забезпечити більш ефективні обчислення.

Наприклад, для експерименту LHC ALICE, grid MC production LHC16a2c2 (p-p, 13 TeV, PHOJET for runs near the VdM scan (new), LHC15h anchor, ALIROOT-6470) було згенеровано 9 900 400 подій, і статистичні дані показують нам, це еквівалентно до 6 CPU рокам 358 CPU дням часу виконання налаштування симуляції, використовуючи та генеруючи 7.457 TB даних. Новий підхід може дозволити зекономити тут принаймні 15000 годин процесора.

Загальна ідея оптимізації полягає в тому, що наш алгоритм може працювати в режимі налаштування, лише використовуючи отримані в процесі оптимізації значення параметрів для аналізу Парето-фронт. Це дозволяє отримати максимальну користь від оптимізації процесу симуляцій.

В даний час ми маємо три сценарії, які слід розглянути:

- алгоритм для гетерогенного кластера (рисунок 4.13);
- алгоритм для вузлів маркованого вручну негетерогенного кластеру (рисунок 4.14);
- алгоритм для випадку неоднорідного кластера (рисунок 4.15).

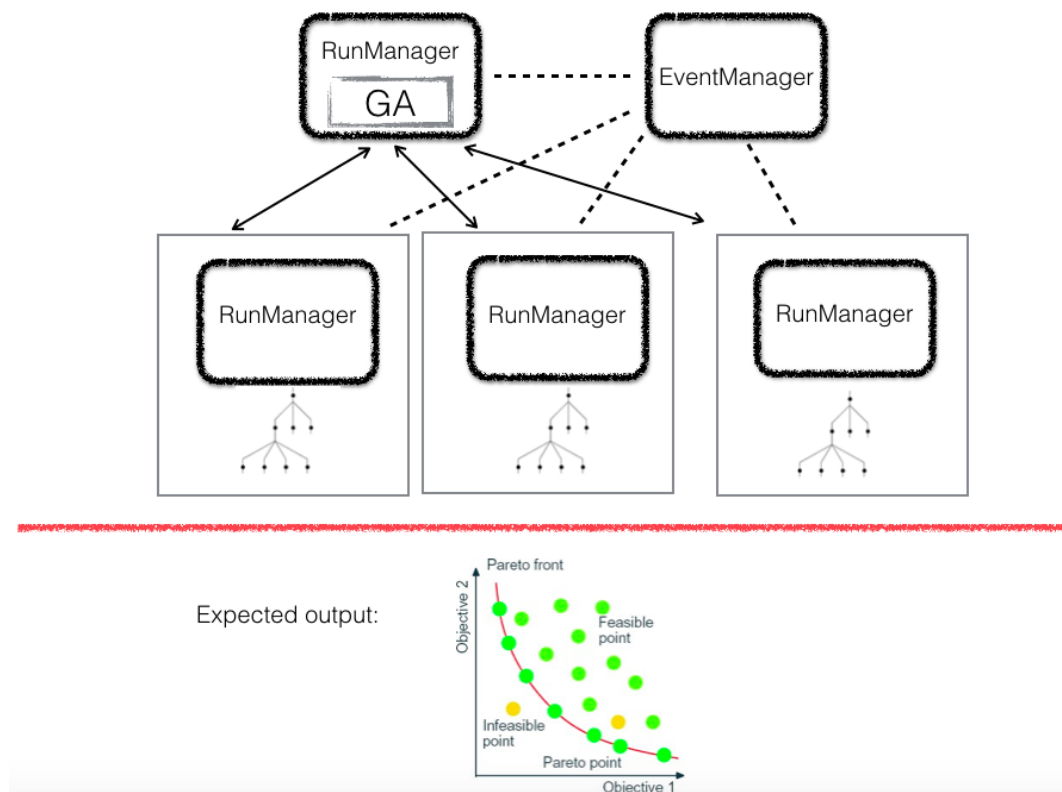


Рисунок 4.13 – Алгоритм для гетерогенного кластера

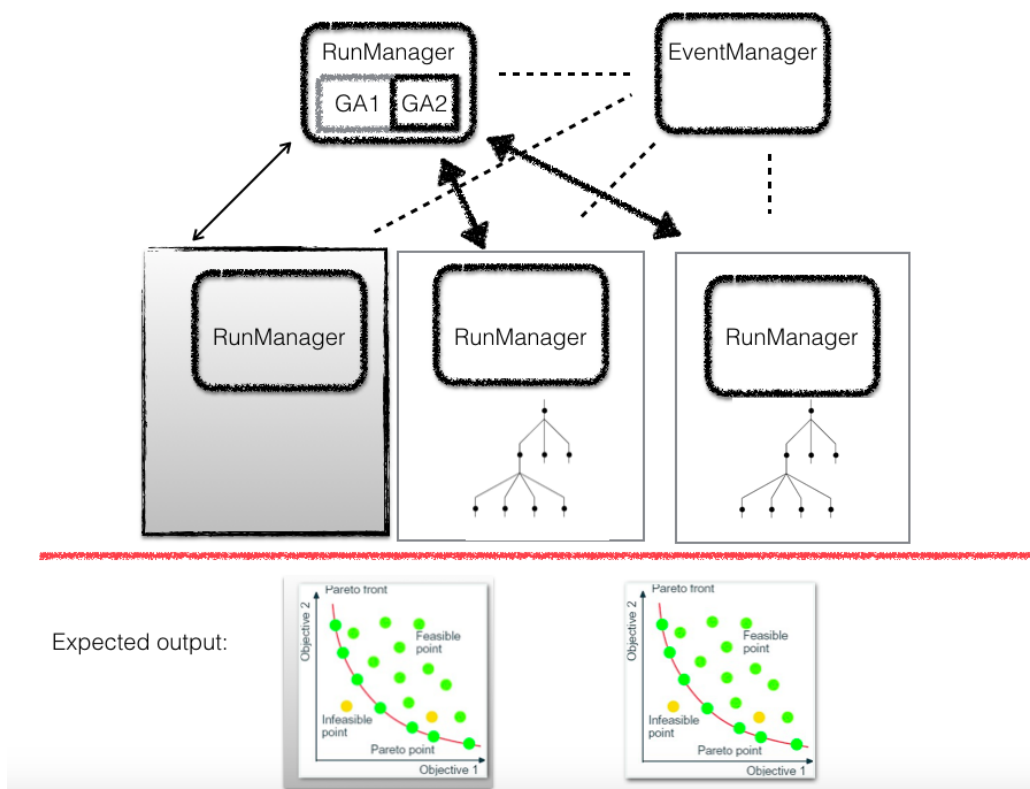


Рисунок 4.14 – Алгоритм для вузлів маркованого вручну негетерогенного

кластера

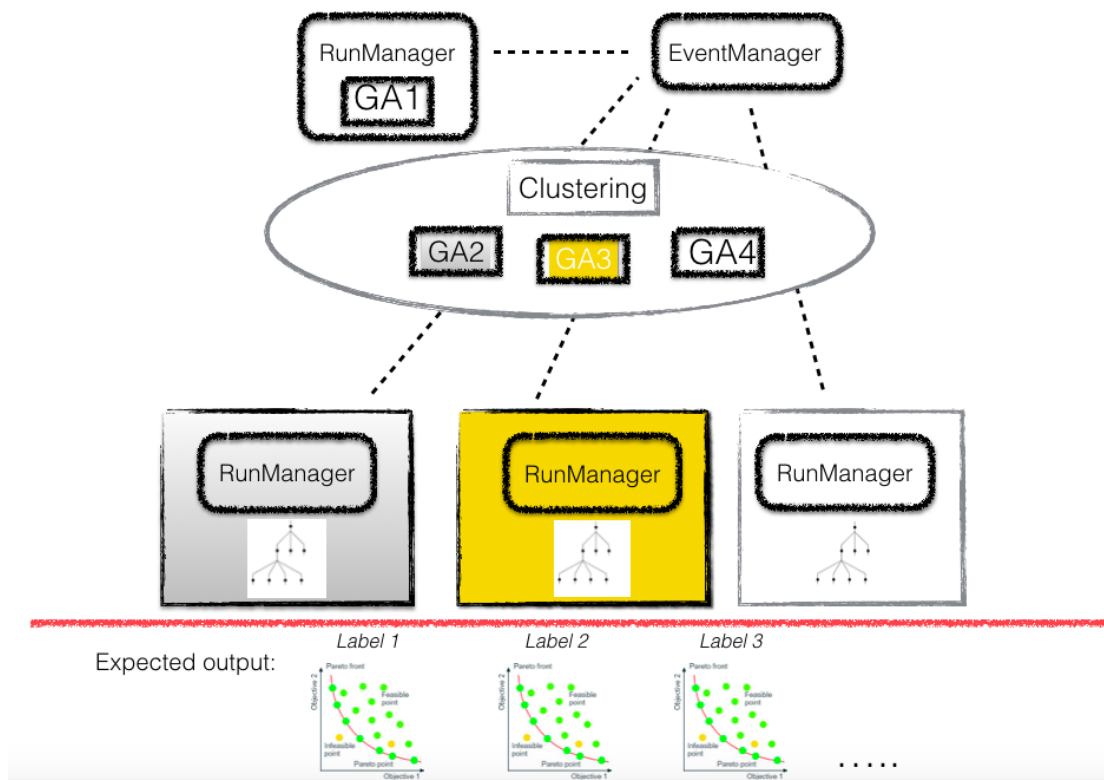


Рисунок 4.15 – Алгоритм для випадку неоднорідного кластера

Стохастична оптимізація базується на налаштуванні параметрів генетичного алгоритму, щоб мати можливість оптимізувати параметри у середовищі для «ідеальних» високопродуктивних обчислень, де всі обчислювальні процеси є «ідеальними» та мають однакову конфігурацію апаратного забезпечення. У генетичному алгоритмі ми використовуємо ідею попередньої обробки даних (параметрів), використовуючи метод головних компонент для очищення зашумленого набору даних, за допомогою введенного нами нового генетичного оператора, що допомагає підвищити конвергенцію генетичного алгоритму.

Основні підходи, що використані в дослідженні оптимізації програмного пакета GeantV:

- з точки зору генетичного алгоритму генерація нашого генетичного

алгоритму становить N завдань;

- генетичний індивідум є вектором вхідних параметрів (у випадку першого покоління це випадково згенеровані обмежені значення);
- функція оцінки функції фітнесу – це вектор значень заданих функцій.

Повторне надсилання нових N завдань з попередньо визначеними генетичними індивідуумами (векторами параметрів), які були отримані за допомогою генетичних еволюційних операторів продовжується до тих пір поки ми не згенеруємо нове покоління. Процес буде знову повторений, поки не буде знайдено «ідеального» налаштування параметрів або ідеальну Парето точку на Парето-фронту. Переваги цієї схеми полягають в тому, що вихідні дані або фізичні результати насправді не змінюється при зміні налаштувань нашої системи, але ми можемо збирати дані про продуктивність системи та спробувати налаштовувати її під час роботи величезної кількості завдань при аналізі даних.

Недолік схеми полягає в тому, що насправді обчислювальні ресурси не настільки ідеальні та при появі нових архітектур, процесорів, графічних процесорів тощо, нам потрібно застосовувати наш метод також для не гетерогенних ресурсів. Ми можемо намагатися бути «експертами» і розподілити ресурси вручну (кажучи, що ми очікуємо, що у нас є « N -груп» машин і організовуємо для кожної групи, окрему процедуру налаштування через « N » генетичних алгоритмів). Це може бути легко виконано в сучасних кластерах, де організовано різні черги для різних типів машин, спираючись на «експертні» знання. Або ми можемо спробувати організувати таку складну систему разом, намагаючись зібрати та реорганізувати процес збору та аналізу даних про характеристики наявних машин у пулі. Цю задачу можна визначити як задачу балансування системи завантаження з урахуванням продуктивності для високопродуктивних обчислень за допомогою зв'язкових наборів даних.

4.5 Тестові налаштунки GeantV

Важливою частиною визначення коректної процедури оптимізації є наявність тестового набору, що охоплює різні випадки використання GeantV, забезпечуючи належне охоплення функціональності та взаємодії між ядром GeantV та іншими його підмодулями: фізикою, геометрією тощо.

Для тестових налаштувань алгоритму оптимізації ми використовували код Run.C, який ще називається прикладом ExN03 (рисунок 4.16), і є прикладом на основі вибірки калориметра з шарами поглинача Pb та розчинів рідини Ar. Всі електромагнітні процеси та розпади моделюються окремими процесами для генерації γ , e^+ , e^- частинок (використовується для вивчення душі частинок).

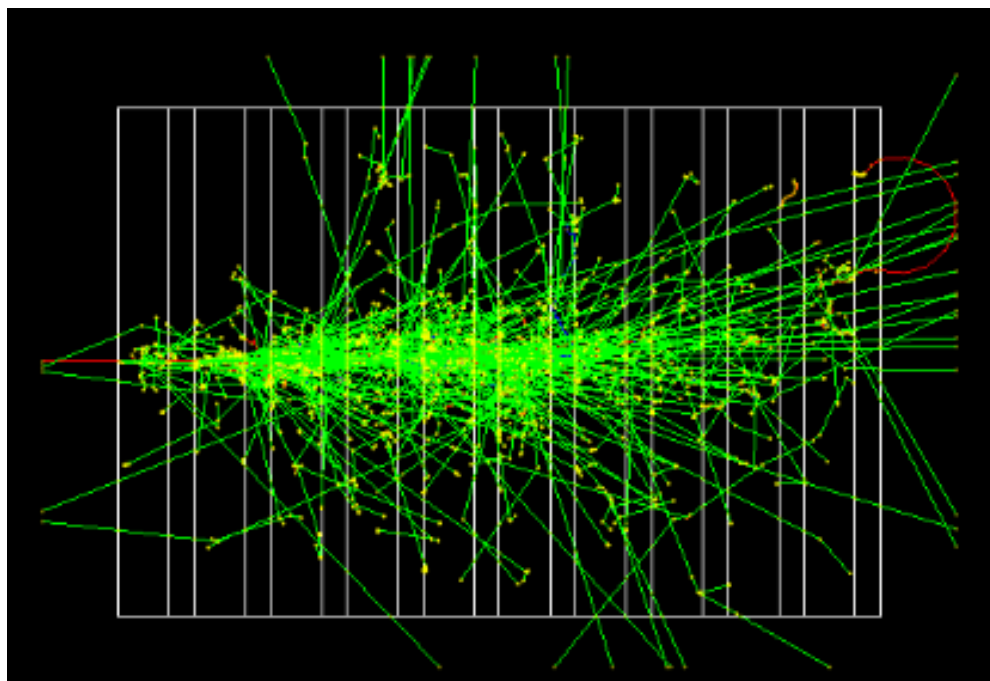


Рисунок 4.16 – Моделювання ExN03 - розробка електромагнітного душі частинок в «сендвіч»-калориметру

В якості результату симуляції ми збираємо дані, зв'язані з відповіддю детектора: зіткнення включають в себе дані про депозит енергії, довжину траєкторії в поглиначі та депозит енергії. Перші результати для прикладу

ExampleN03 показують дуже оптимістичний ефект використання нашого підходу для оптимізації з точки зору поліпшення обчислювальної продуктивності моделювання GeantV.

4.6 Результати тестування

Вимірювання зроблені під час запуску моделювання ExN03 прикладу на рисунку 4.17, 4.18, 4.19, 4.20 та 4.21 показують цікаві результати з точки зору поліпшення обчислювальної продуктивності моделювання GeantV.

На рисунку 4.17, 4.18 зображені результати для моделювання ExN03 та перші результати оптимізації. Для платформи Intel(R) Xeon(R) CPU E5-2695 - Ivy Bridge можна помітити значне покращення, як для значень метрики що вимірює час обробки симуляції (англ. Run Time (RT)) так і по значенням метрики ефективного використання апаратного часу (англ. CPU Time). Також для платформи Intel(R) Xeon(R) CPU E5-2695 – Ivy Bridge можна помітити значне покращення значень метрики, яка вимірює ефективність роботи симуляції, а саме кількість оброблених зіткнень за одиницю часу (англ. primaries per time) та яка є найбільш цікавою для нас.

На рисунку 4.17, 4.18 зображені розширені результати для моделювання ExN03 для платформи Intel® Core™ i7-8700K. Тут включені розширені метричні вимірювання, які описують ефективність коду та його ефективність, такі як метрика для вимірювання кількості використаних циклів процесора (англ. - number of CPU cycles), метрика для вимірювання кількості всіх циклів корисної роботи (англ. – number of cycles), метрика для вимірювання кількості використаних інструкцій (англ. number of instructions), метрика для вимірювання кількості відгалужень у коді (англ. number of branches), метрика для вимірювання кількості помилок у відгалуженнях у коді (англ. number of the branch misses), метрика для вимірювання пропущених при обробці даних

кешу у кеш-пам'яті L1d (англ. number of L1d cache misses), метрика для вимірювання пропущених даних кешу у кеш-пам'яті L1i (англ. number of L1i cache misses). Всі ці метрики показують значне покращення для оптимізованої версії коду.

Як і для платформи Intel(R) Xeon(R) CPU E5-2695 – Ivy Bridge, для платформи Intel® Core™ i7-8700K також можна помітити значне покращення значень метрики, яка вимірює ефективність роботи симуляції, а саме кількість оброблених зіткнень за одиницю часу (Primitives per time) та яка є найбільш цікавою для нас.

Попередні результати показують, що стохастично налаштовані параметри для паралельних додатків GeantV на обчислювальних процесорах Intel® Core™ i7-8700K (рисунок 4.19, 4.20 та 4.21) змогли отримати перевагу від 18% прискорення роботи у порівнянні з початковим набором. Тим часом результати, отримані для процесора Intel (R) Xeon (R) CPU E5-2695 (рисунок 4.17, 4.18), показують, що, налаштувавши ці параметри, ми змогли досягти зниження частки використання ЦП до 34%, що забезпечує стабільне використання пам'яті та зменшення пробігу час роботи пакетної роботи до 27%.

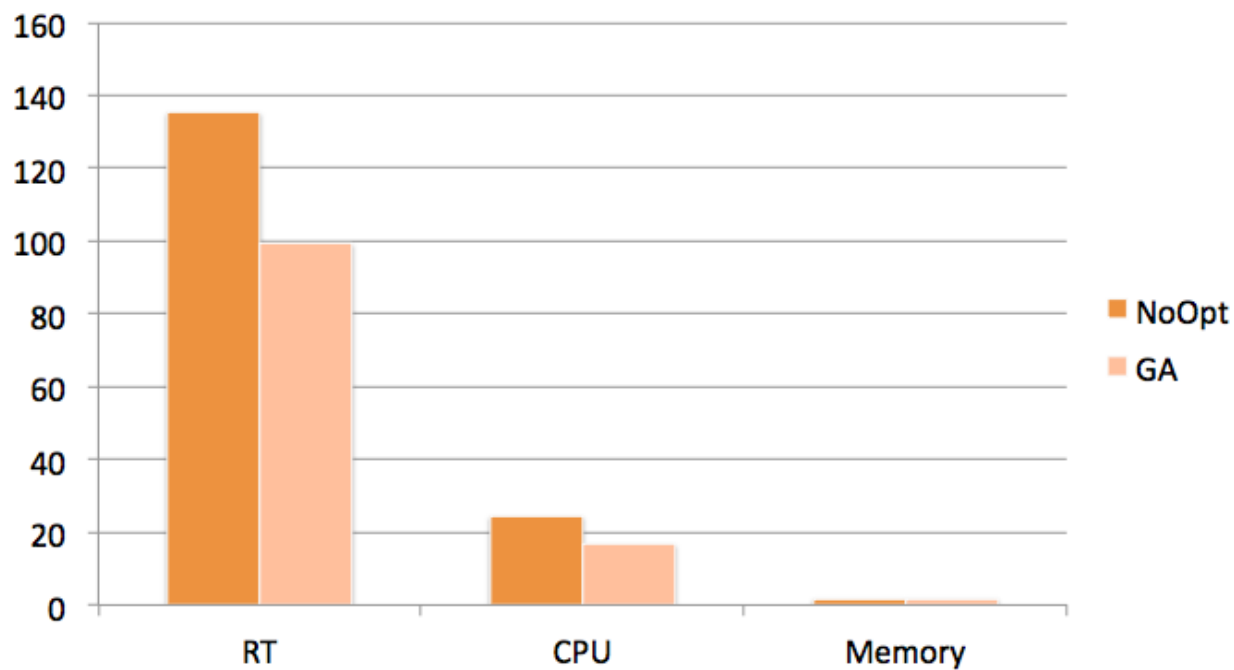


Рисунок 4.17 – Перші результати оптимізації (Intel(R) Xeon(R) CPU E5-2695 - Ivy Bridge) для моделювання ExN03

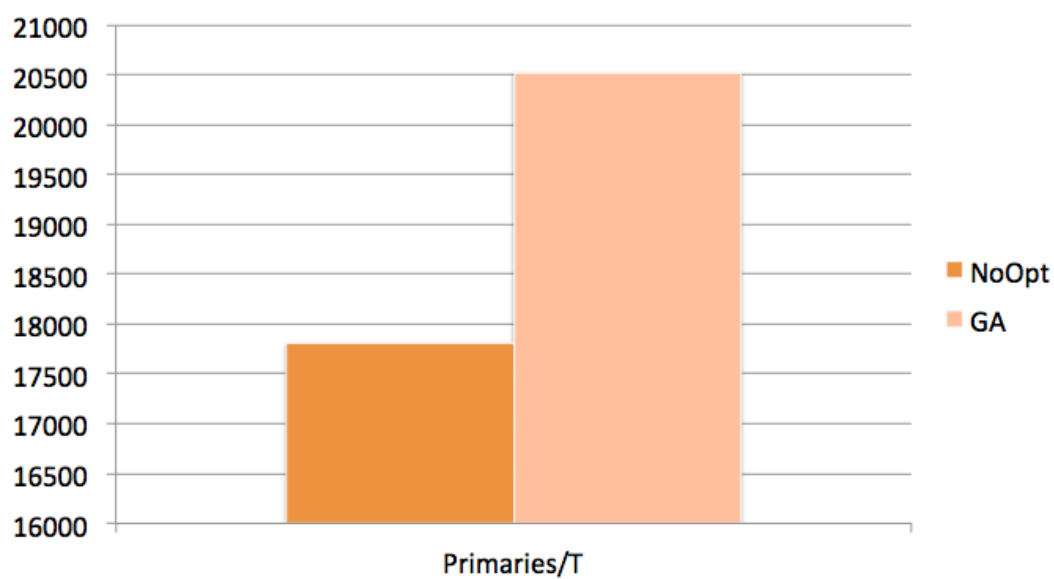


Рисунок 4.18 – Перші результати оптимізації (Intel(R) Xeon(R) CPU E5-2695 - Ivy Bridge) для моделювання ExN03

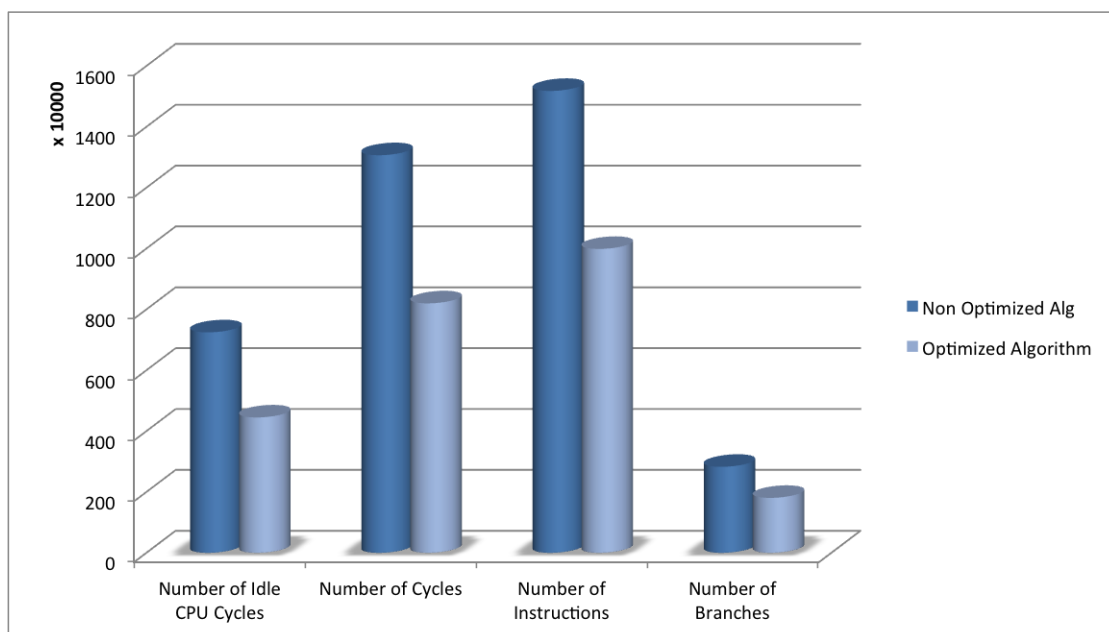


Рисунок 4.19 – Перші результати оптимізації (Intel® Core™ i7-8700K) для моделювання ExN03

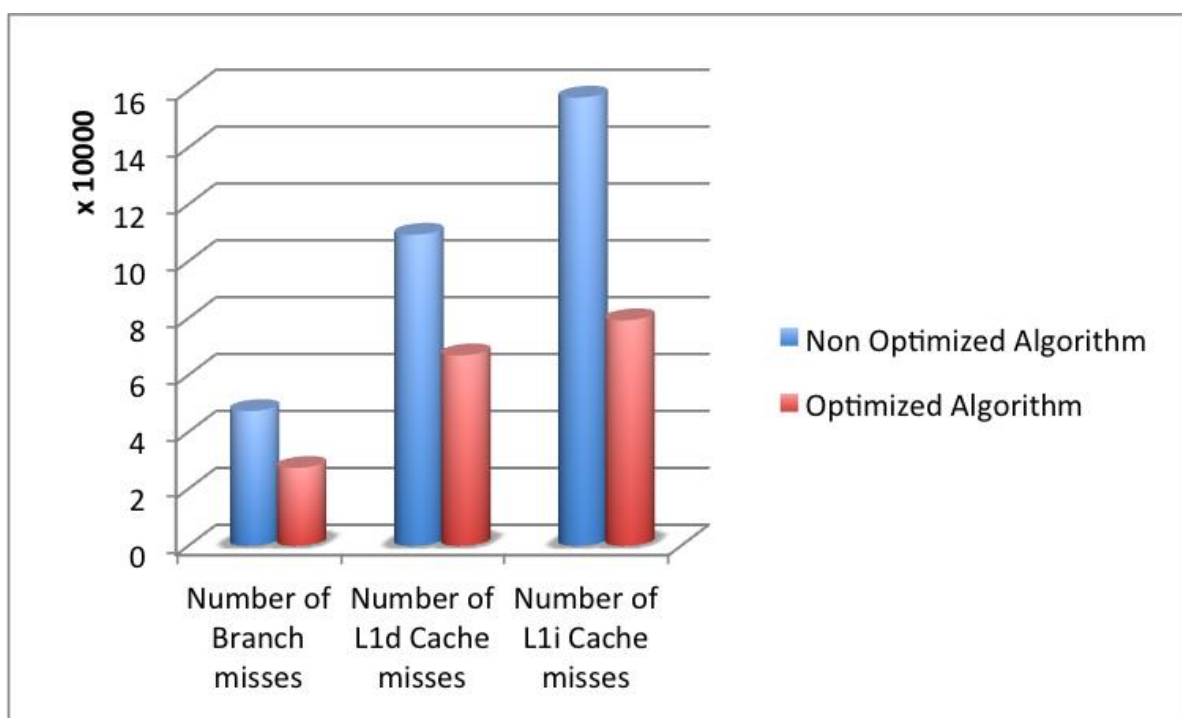


Рисунок 4.20 – Перші результати оптимізації (Intel® Core™ i7-8700K) для моделювання ExN03

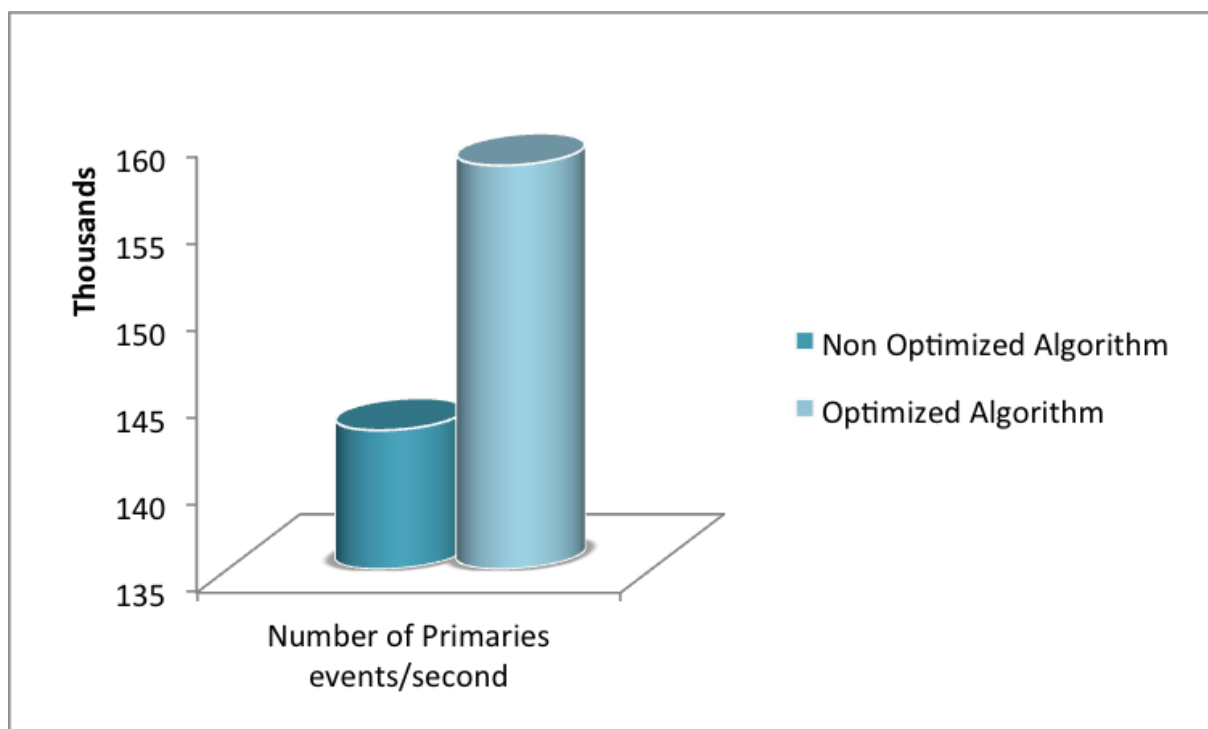


Рисунок 4.21 – Перші результати оптимізації (Intel® Core™ i7-8700K) для моделювання ExN03

Висновки до розділу 4

– Виконано інтеграцію НГК-оператора в типовий генетичний алгоритм, що використовується для оптимізації програмного пакета GeantV, і проведено дослідження його продуктивності в порівнянні з застосуванням немодифікованого генетичного алгоритму [105], [91]. Дослідження показали, що для еталонних тестів багатокритеріальних задач оптимізації загальний час виконання батчу симуляцій за допомогою модифікованого генетичного алгоритму скоротився на 18% (на обчислюваних процесорах Intel® Core™ i7-8700K) і на 27% з одночасним зменшенням часу використання CPU на 34% (для процесорів Intel(R) Xeon(R) CPUE5-2695).

– Результати, що отримані в ході цього дослідження, є доказом ефективності застосування концепції оптимізації продуктивності GeantV з використанням методів еволюційного налаштування обчислень. Цей же метод

може бути застосований для оптимального встановлення GeantV-додатків на суперкомп'ютерах або кластерах для високопродуктивних обчислень, одночасно з можливістю налаштування високо паралельних задач, оброблених у неоднорідному середовищі, для досягнення масштабованості в середовищах для високопродуктивних обчислень.

Результати досліджень за четвертим розділом опубліковані в статтях:

1. Оксана Шадура. Метод головних компонент і оптимізація пакетів фізичного моделювання за допомогою генетичних алгоритмів // Вісник Університету «Україна», Серія «Інформатика, обчислювальна техніка та кібернетика». – 2019. – №1 (22) – С. 198–209. – Бібліогр.: 16 назв.

2. Оксана Шадура. Оптимізація пакету фізичного моделювання GeantV за допомогою генетичних алгоритмів. // Вчені записки Таврійського національного університету імені В. І. Вернадського, Серія: Технічні науки – 2019. – Том 30 (69). – № 3. – С. 206–213.

ВИСНОВКИ

Дослідження, виконані під час роботи над кандидатською дисертацією, були спрямовані на розв'язання задачі стохастичної оптимізації для паралельних додатків для моделювання транспорту частинок у складних детекторах. У ході роботи був представлений новий генетичний оператор, здатний прискорити конвергенцію алгоритму до справжнього Парето-фронт, проведено аналіз власних значень для нового алгоритму та спостереження за спеціальними точками (фіксованими точками) для генетичного алгоритму, також була проведена робота над введенням схеми балансування подій для розподіленого робочого навантаження для додатка GeantV.

Результати впровадження генетичного оператора дозволяють прискорити конвергенцію у два рази, що дозволяє швидше отримати доступ до раннього Парето-фронт.

1. Попередні результати показують, що стохастично налаштовані параметри для паралельних додатків GeantV на обчислюваних процесорах Intel® Core™ i7-8700K змогли отримати перевагу від 18 % прискорення партії роботи у порівнянні з початковим набором. Тим часом результати, отримані для процесора Intel (R) Xeon (R) CPU E5-2695, показують, що, налаштовувавши ці параметри, ми змогли досягти зниження частки використання ЦП до 34 %, що забезпечує стабільне використання пам'яті та зменшення пробігу час роботи пакетної роботи до 27 %.

2. Результати, отримані в ході цього дослідження, є доказом концепції оптимізації продуктивності GeantV з використанням еволюційних налаштувань для обчислень. Цей же метод може бути застосований для встановлення оптимально GeantV-додатків на суперкомп'ютерах і кластерах для високопродуктивних обчислень, одночасно з можливістю налаштування масово паралельних наборів задач, оброблених у неоднорідному середовищі, і для досягнення масштабованості у середовищах

для високопродуктивних обчислень.

3. Проведено аналіз програмного забезпечення, яке використовується для симуляції (моделювання) проходження елементарних частинок через матерію при обробці експериментів в ЦЕРН на Великому адронному колайдері, зокрема, програмного пакета нового покоління GeantV. Аналіз дозволив визначити головні чинники (розмір пам'яті, обсяг даних, час роботи, кількість інструкцій як фітнес-функції та кількість фізичних подій, кількість фізичних подій в буфері, кількість процесів та кількість препаративів), що впливають на продуктивність обчислень та можуть бути використані для стохастичної оптимізації продуктивності пакету симуляції, а також виявити вузькі місця в функціональній моделі GeantV.

4. Сформульована багатоцільова задача оптимізації продуктивності пакету симуляції транспорту частинок GeantV, обґрунтовано вибір генетичного алгоритму в якості базової процедури такої оптимізації та визначені простір рішень та цільовий простір. Використовуючи цей аналіз, представлення генетичного алгоритму як динамічної системи через його зв'язок з марківськими ланцюжками та аналіз властивостей власних чисел матриці переходу, вдалося визначити оцінки Парето-фронту, оптимальної Парето точки та нерухомої точки.

5. Розроблено метод незцентрованих головних компонент, сформульовані його математичні основи та отримані формули для оцінки наближень при аналізі даних, які забезпечують контролювання збереження максимуму інформації при цих наближеннях. Запропоновано підходи що до включення цього методу в генетичний алгоритм через введення нового оператора в динамічну систему генетичного алгоритму.

6. Проведено шляхом моделювання порівняльний аналіз використання різних варіантів включення методу головних компонент в генетичний алгоритм (NSGA-II, NSGA-II у комбінації з оператором UPCA1, NSGA-II у комбінації з оператором UPCA2, NSGA-II у комбінації з оператором UPCA3,

NSGA-II у комбінації з оператором UPCA4, NSGA-II у комбінації з оператором UPCA5 і NSGA-II у комбінації з оператором UPCA1) для центрованих і незцентрованих даних за допомогою оцінки середньоквадратичної похибки і показано, що NSGA-II у комбінації з UPCA4 є найбільш ефективним для конвергенції алгоритму до Парето-фронт.

7. Вперше запропоновано новий генетичний оператор (НГК-оператор), який побудовано на основі методу незцентрованих головних компонент, і показано, що його застосування в тестових багатоцільових задачах оптимізації забезпечує прискорення збіжності генетичного алгоритму до оптимального Парето-фронт.

8. Виконано інтеграцію НГК-оператора в типовий генетичний алгоритм, що використовується для оптимізації програмного пакета GeantV, і проведено дослідження його продуктивності в порівнянні з застосуванням немодифікованого генетичного алгоритму. Дослідження показали, що для еталонних тестів багатоцільових задач оптимізації загальний час виконання батчу симуляцій за допомогою модифікованого генетичного алгоритму скоротився на 18 % (на обчислюваних процесорах Intel® Core™ i7-8700K) і на 27 % з одночасним зменшенням часу використання CPU на 34 % (для процесорів Intel(R) Xeon(R) CPUE5-2695).

ПЕРЕЛІК ПОСИЛАНЬ

[1] Georges Aad, JM Butterworth, J Thion, U Bratzler, PN Ratoff, RB Nickerson, JM Seixas, I Grabowska-Bold, F Meisel, S Lokwitz, et al. The ATLAS experiment at the CERN Large Hadron Collider. *Jinst*, 3:S08003, 2008.

[2] Kenneth Aamodt, A Abrahantes Quintana, R Achenbach, S Acounis, D Adamová, C Adler, M Aggarwal, F Agnese, G Aglieri Rinella, Z Ahammed, et al. The ALICE experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08002, 2008.

[3] N Abgrall, O Andreeva, A Aduszkiewicz, Yasir Ali, T Anticic, N Antoniou, B Baatar, F Bay, A Blondel, J Blumer, et al. NA61/SHINE facility at the cern sps: beams and detector system. *Journal of Instrumentation*, 9(06):P06005, 2014.

[4] Allison, J., et al. Recent developments in Geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 835 (2016): 186-225.

[5] Sea Agostinelli, John Allison, K al Amako, J Apostolakis, H Araujo, P Arce, M Asai, D Axen, S Banerjee, G Barrand, et al. Geant4 - a simulation toolkit. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[6] C. Ahdida and all. The experimental facility for the Search for Hidden Particles at the CERN SPS. *Journal of Instrumentation*, 14(03):P03025–P03025, mar 2019.

[7] Sergey Alekhin, Wolfgang Altmannshofer, Takehiko Asaka, Brian Batell, Fedor Bezrukov, Kyrylo Bondarenko, Alexey Boyarsky, Ki-Young Choi, Cristobal Corral, Nathaniel Craig, et al. A facility to search for hidden particles at the CERN SPS: the SHIP physics case. *Reports on progress in physics*, 79(12):124201, 2016.

[8] Allison, John. Facilities and methods: Geant4 – a simulation toolkit. *Nuclear Physics News*, 17.2 (2007): 20-24.

[9] Apostolakis, John, et al. Geant4 low energy electromagnetic models for electrons and photons. *No. CERN-OPEN-99-034*. 1999.

[10] Amako, Katsuya, and for the Geant4 Collaboration. Present status of Geant4. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 453.1-2 (2000): 455-460

[11] A Augusto Alves Jr, LM Andrade Filho, AF Barbosa, I Bediaga, G Cernicchiaro, G Guerrier, HP Lima Jr, AA Machado, J Magnin, F Marujo, et al. The LHCb detector at the LHC. *Journal of instrumentation*, 3(08):S08005, 2008.

[12] Mohammad Al-Turany, D Bertini, R Karabowicz, D Kresan, P Malzacher, T Stockmanns, and F Uhlig. The fairroot framework. In *Journal of Physics: Conference Series*, volume 396, page 022001. IOP Publishing, 2012.

[13] G Amadio, A Ananya, J Apostolakis, A Arora, M Bandieramonte, A Bhattacharyya, C Bianchini, R Brun, P Canal, F Carminati, L Duhem, D Elvira, A Gheata, M Gheata, I Goulas, R Iope, S Jun, G Lima, A Mohanty, T Nikitina, M Novak, W Pokorski, A Ribon, R Sehgal, O Shadura, S Vallecorsa, S Wenzel, and Y Zhang. GeantV: from cpu to accelerators. *Journal of Physics: Conference Series*, 762(1):012019, 2016.

[14]. Amadio, Guilherme, et al. Increasing Parallelism in the ROOT I/O Subsystem. *Journal of Physics: Conference Series*. Vol. 1085. No. 3. IOP Publishing, 2018.

[15] G Amadio, J Apostolakis, M Bandieramonte, A Bhattacharyya, C Bianchini, R Brun, Ph Canal, F Carminati, L Duhem, D Elvira, J de Fine Licht, A Gheata, R L Iope, G Lima, A Mohanty, T Nikitina, M Novak, W Pokorski, R Seghal, O Shadura, S Vallecorsa, and S Wenzel. The GeantV project: preparing the future of simulation. *Journal of Physics: Conference Series*, 664(7):072006, 2015.

[16] G Amadio, F Hariri, P Canal, L Duhem, M Gheata, G Cosmo, V Ivantchenko, O Shadura, SP Behera, G Folger, et al. Stochastic optimization of geantv code by use of genetic algorithms. In *J. Phys. Conf. Ser.*, volume 898, page

042026, 2017.

[17] David P Anderson. Boinc: A system for public-resource computing and storage. In *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10. IEEE Computer Society, 2004.

[18] Costas Andreopoulos, A Bell, D Bhattacharya, F Cavanna, J Dobson, S Dytman, H Gallagher, P Guzowski, R Hatcher, P Kehayias, et al. The genie neutrino monte carlo generator. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 614(1):87–104, 2010.

[19] Giorgio Apollinari, I Béjar Alonso, Oliver Brüning, M Lamont, and Lucio Rossi. High-luminosity large hadron collider (HL-LHC): Preliminary design report. *Technical report, Fermi National Accelerator Lab.(FNAL)*, Batavia, IL (United States), 2015.

[20] J Apostolakis, M Bandieramonte, G Bitzes, R Brun, P Canal, F Carminati, G Cosmo, J C De Fine Licht, L Duhem, V D Elvira, A Gheata, S Y Jun, G Lima, T Nikitina, M Novak, R Sehgal, O Shadura, and S Wenzel. Towards a high performance geometry library for particle-detector simulations. *Journal of Physics: Conference Series*, 608(1):012023, 2015.

[21] Apostolakis, John, et al. A concurrent vector-based steering framework for particle transport. *Journal of Physics: Conference Series*. Vol. 523. No. 1. IOP Publishing, 2014.

[22] Apostolakis, John, et al. Rethinking particle transport in the many-core era towards GEANT 5. *Journal of Physics: Conference Series*. Vol. 396. No. 2. IOP Publishing, 2012.

[23] J Apostolakis, M Bandieramonte, G Bitzes, R Brun, P Canal, F Carminati, JC De Fine Licht, L Duhem, VD Elvira, A Gheata, et al. Adaptive track scheduling to optimize concurrency and vectorization in geantv. In *Journal of Physics: Conference Series*, volume 608, page 012003. IOP Publishing, 2015.

[24] P Baudreghien, A Macpherson, R Calaga, V Parma, E Jensen, E

Montesinos, O Capatina, and K Brodzinski. Functional specifications of the LHC prototype crab cavity system. Technical report, 2013.

[25] Daniele Bonacorsi and Tiziana Ferrari. WLCG service challenges and tiered architecture in the LHC era. In *IFAE 2006*, pages 365–368. Springer, 2007.

[26] D. Brockhoff and E. Zitzler. Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. *Evolutionary Computation*, 17(2):135–166, 2009.

[27] François Broquedis, Jérôme Clet-Ortega, Stéphanie Moreaud, Nathalie Furmento, Brice Goglin, Guillaume Mercier, Samuel Thibault, and Raymond Namyst. hwloc: A generic framework for managing hardware affinities in hpc applications. In *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pages 180–186. IEEE, 2010.

[28] Rene Brun and Fons Rademakers. ROOT - an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1-2):81–86, 1997.

[29] Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4(1):41–59, 1977.

[30] P Charpentier. Benchmarking worker nodes using LHCb productions and comparing with HEPSpec06. *Journal of Physics: Conference Series*, 898:082011, oct 2017.

[31] Adolphi, Roman. The CMS experiment at the CERN LHC. *JINST*, (2008): S08004.

[32] Alexandre d’Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[33] Lawrence Davis. Handbook of Genetic Algorithms, *Van Nostrand Reinhold*, New York, 1991.

[34] Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated binary

crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[35] Kalyanmoy Deb and Debayan Deb. Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, 4(1):1–28, 2014.

[36] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4):577–601, 2014.

[37] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[38] Kalyanmoy Deb and D Saxena. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360, 2006.

[39] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. In *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*, volume 1, pages 825–830. IEEE, 2002.

[40] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. *Scalable Test Problems for Evolutionary Multiobjective Optimization*, pages 105–145. Springer London, London, 2005.

[41] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer, 2011.

[42] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr 2002.

[43] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y

Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[44] Agoston E Eiben, P-E Raue, and Zs Ruttkay. Genetic algorithms with multi-parent recombination. In *International Conference on Parallel Problem Solving from Nature*, pages 78–87. Springer, 1994.

[45] The LEP Working Group for Higgs, ALEPH Collaboration, DELPHI Collaboration, L3 Collaboration, OPAL Collaboration, et al. Search for the standard model higgs boson at lep. *Physics Letters B*, 565:61–75, 2003.

[46] Hilda Geiringer. On the probability theory of linkage in mendelian heredity. *The Annals of Mathematical Statistics*, 15(1):25–57, 1944.

[47] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[48] E Cortina Gil, E Martn Albarrán, E Minucci, G Nüssle, S Padolski, P Petrov, N Szilasi, B Velghe, G Georgiev, V Kozhuharov, et al. The beam and detector of the na62 experiment at cern. *Journal of Instrumentation*, 12(05):P05025, 2017.

[49] David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989:102, 1989.

[50] David E Goldberg and Philip Segrest. Finite markov chain analysis of genetic algorithms. In *Proceedings of the second international conference on genetic algorithms*, volume 1, page 1, 1987.

[51] Mickael Goujon, Hamish McWilliam, Weizhong Li, Franck Valentin, Silvano Squizzato, Juri Paern, and Rodrigo Lopez. A new bioinformatics analysis tools framework at embl–ebi. *Nucleic acids research*, 38(suppl_2):W695–W699, 2010.

[52] Andrzej Granas and James Dugundji. *Fixed point theory*. Springer Science & Business Media, 2013.

[53] Elena Graverini, Thomas Ruf, and Eric Van Herwijnen. Mass dependence of branching ratios into HNL for FairShip. *Technical report*, CERN,

2016.

[54] John J Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on systems, man, and cybernetics*, 16(1):122–128, 1986.

[55] LHC Study Group and European Organization for Nuclear Research. *Design study of the Large Hadron Collider (LHC): a multiparticle collider in the LEP tunnel*. Number 3. Not Avail, 1991.

[56] Durillo, Juan J., and Antonio J. Nebro. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* 42.10 (2011): 760-771.

[57] Nikolaus Hansen, Dirk V. Arnold, and Anne Auger. Evolution Strategies. In Janusz Kacprzyk and Witold Pedrycz, editors, *Handbook of Computational Intelligence*. 2015.

[58] Christina Hayes and Tomáš Gedeon. Hyperbolicity of the fixed point set for the simple genetic algorithm. *Theoretical Computer Science*, 411(25):2368–2383, 2010.

[59] John H Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. 1975.

[60] Paul Honeine. An eigenanalysis of data centering in machine learning. *arXiv preprint arXiv:1407.2904*, 2014.

[61] Holger H Hoos and Thomas Stützle. *Stochastic local search: Foundations & applications*. Elsevier, 2004.

[62] Holger H. Hoos and Thomas Stutzle. *Stochastic Local Search Algorithms: An Overview*, pages 1085–1105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[63] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[64] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A

scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer, 2005.

[65] V.N. Ivanchenko for Geant4 Collaboration, et al. Geant4 toolkit for simulation of hep experiments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 502(2):666–668, 2003.

[66] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans. Evolutionary Computation*, 18(4):602–622, 2014.

[67] Sverre Jarp, Alfio Lazzaro, and Andrzej Nowak. The future of commodity computing and many-core versus the interests of HEP software. In *Journal of Physics: Conference Series*, volume 396, page 052058. IOP Publishing, 2012.

[68] J.Cadima and I.Jolliffe. On relationships between uncentered and column-centered principal component analysis. *Pak.J.Statist.*, 2009.

[69] Jolliffe, Ian. Principal component analysis. *Springer Berlin Heidelberg*, 2011.

[70] John G Kemeny, James Laurie Snell, et al. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.

[71] Mohd Ehmer Khan, Farmeena Khan, et al. A comparative study of white box, black box and grey box testing techniques. *Int. J. Adv. Comput. Sci. Appl*, 3(6), 2012.

[72] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[73] John R Koza. Genetic programming. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.5416>, 1997.

[74] Alexander v. d. Kuijl, Michael T. M. Emmerich, and Hui Li. A robust multi-objective resource allocation scheme incorporating uncertainty and service

differentiation. *Concurrency and Computation: Practice and Experience*, 22(3):314–328, 2010.

[75] Marco Laumanns, Eckart Zitzler, and Lothar Thiele. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 181–196. Springer, 2001.

[76] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[77] Daniel Lombrana Gonzalez, Francois Grey, Jakob Blomer, Predrag Buncic, Artem Harutyunyan, Miguel Marquina, Ben Segal, Peter Skands, and Anton Karneyeu. Virtual machines & volunteer computing: Experience from lhc@home: Test4theory project. In *The International Symposium on Grids and Clouds (ISGC) 2012*, volume 153, page 036. SISSA Medialab, 2012.

[78] D Lombraña González, A Harutyunyan, B Segal, I Zacharov, E McIntosh, PL Jones, M Giovannozzi, L Rivkin, MA Marquina, P Skands, et al. Lhc@ home: a volunteer computing system for massive numerical simulations of beam dynamics and high energy physics events. In *Conf. Proc.*, volume 1205201, pages 505–507, 2012.

[79] Antonio López Jaimes, Carlos A Coello Coello, and Debrup Chakraborty. Objective reduction using a feature selection technique. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 673–680. ACM, 2008.

[80] Sean Luke and Lee Spector. A comparison of crossover and mutation in genetic programming. *Genetic Programming*, 97:240–248, 1997.

[81] Clifford Lynch. Big data: How do your data grow? *Nature*, 455(7209):28, 2008.

[82] Edoardo Martelli and S Stancu. LHCCOPN and LHCONE: Status and future evolution. In *Journal of Physics: Conference Series*, volume 664, page

052025. IOP Publishing, 2015.

[83] Kurt Marti. *Stochastic optimization methods*, Volume 2. Springer, 2005.

[84] Boris S Mitavskiy, Elio Tuci, Chris Cannings, Jonathan Rowe, and Jun He. Geiringer theorems: from population genetics to computational intelligence, memory evolutive systems and hebbian learning. *Natural Computing*, 12(4):473–484, 2013.

[85] B. Mitavskiy and J. Rowe. An extension of geiringer’s theorem for a wide class of evolutionary search algorithms. *Evolutionary Computation*, 14(1):87–118, March 2006.

[86] Sushmita Mitra and Haider Banka. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477, 2006.

[87] Oscar Montiel, Oscar Castillo, Patricia Melin, and Roberto Sepúlveda. Black box evolutionary mathematical modeling applied to linear systems. *Int. J. Intell. Syst.*, 20(2):293–311, 2005.

[88] Sandro Wenzel, Axel Naumann. C++ needs language support for vectorization, *ISO/IEC JTC1 SC22 WG21 N3774*. (2013).

[89] Shadura, Oksana & Petrenko, Anatoly & Svistunov, Sergiy. Multivariate convergence-targeted operator for the genetic algorithm. *System research and information technologies*, P.126-140. 2017. 10.20535/SRIT.2308-8893.2017.4.10.

[90] Шадура, О. Модифікація генетичних алгоритмів на основі методу нецетрованих головних компонент та стандартні тести. World Science, № 4 (44), стр. 4–11, DOI: 10.31435/rsglobal_ws (2019).

[91] Шадура, О. Оптимізація пакету фізичного моделювання GeantV за допомогою генетичних алгоритмів. Вчені записки Таврійського національного університету імені В. І. Вернадського, Серія: Технічні науки, т. 30 (69), № 3, стр. 206–213 (2019).

[92] Шадура, О. Метод головних компонент і оптимізація пакетів фізичного моделювання за допомогою генетичних алгоритмів. Вісник Університету «Україна», Серія: Інформатика, обчислювальна техніка та кібернетика, №1 (22), стр. 198-209 (2019).

[93] De Weck. O. Multiobjective optimization: History and promise. *In: Proceedings of the Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan, 2004.*

[94] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[95] Riccardo Poli and William B Langdon. Schema theory for genetic programming with one-point crossover and point mutation. *Evolutionary Computation*, 6(3):231–252, 1998.

[96] Riccardo Poli, Christopher R Stephens, Alden H Wright, and Jonathan E Rowe. A schema-theory-based extension of geiringer’s theorem for linear gp and variable-length gas under homologous crossover. In *FOGA*, pages 45–62, 2002.

[97] Colin R. Reeves and Jonathan E. Rowe. *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[98] Jonathan E. Rowe. Genetic algorithm theory. In *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO ’11*, pages 1029–1052, New York, NY, USA, 2011. ACM.

[99] Reeves, Colin, and Jonathan E. Rowe. Genetic algorithms: principles and perspectives: a guide to GA theory. *Springer Science & Business Media*, Vol. 20, 2002.

[100] Günter Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE transactions on neural networks*, 5(1):96–101, 1994.

[101] Frank Schmidt. Sixtrack: version 1, single particle tracking code treating transverse motion with synchrotron oscillations in a symplectic manner.

Technical report, CM-P00049314, 1990.

[102] Lothar M Schmitt. Theory of genetic algorithms. *Theoretical Computer Science*, 259(1):1–61, 2001.

[103] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[104] Oksana Shadura and Federico Carminati. Stochastic performance tuning of complex simulation applications using unsupervised machine learning. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[105] O Shadura, F Carminati, and A Petrenko. Performance optimization of physics simulations through genetic algorithms. *Journal of Computer Science*, 15:57–66, 01 2019.

[106] Jamie Shiers. The worldwide lhc computing grid (worldwide lcg). *Computer physics communications*, 177(1-2):219–223, 2007.

[107] Hemant Kumar Singh, Amitay Isaacs, and Tapabrata Ray. A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 15(4):539–556, 2011.

[108] Torbjorn Sjostrand, Stephen Mrenna, Emanuel Norrbin, Christer Friberg, Patrik Eden, Leif Lonnblad, and Gabriela Miu. High-energy physics event generation with pythia 6.1. *Comput. Phys. Commun.*, 135(LU-TP-00-30):238–259, 2000.

[109] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A brief introduction to pythia 8.1. *Computer Physics Communications*, 178(11):852–867, 2008.

[110] Avinash Sodani. Knights landing (knl): 2nd generation intel xeon phi processor. In *2015 IEEE Hot Chips 27 Symposium (HCS)*, pages 1–24. IEEE, 2015.

[111] Joe Suzuki. A markov chain analysis on simple genetic algorithms.

IEEE Transactions on Systems, Man, and Cybernetics, 25(4):655–659, 1995.

[112] Luke Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[113] David A Van Veldhuizen and Gary B Lamont. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, pages 221–228, 1998.

[114] M.D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. A Bradford book. MIT Press, 1999.

[115] Marty Wattenberg and Ari Juels. Stochastic hillclimbing as a baseline method for. Technical report, Berkeley, CA, USA, 1994.

[116] W. Wenzel and K. Hamacher. Stochastic tunneling approach for global minimization of complex potential energy landscapes. *Phys. Rev. Lett.*, 82:3003–3007, Apr 1999.

[117] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[118] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[119] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[120] Xin-She Yang. Swarm intelligence based algorithms: a critical analysis. *Evolutionary Intelligence*, 7(1):17–28, 2014.

[121] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

Додаток А

СХЕМА АЛГОРИТМУ NSGA-II

```

Input:  $Population_{size}$ , ProblemSize,  $P_{crossover}$ ,  $P_{mutation}$ 
Output: Children
Population  $\leftarrow$  InitializePopulation( $Population_{size}$ , ProblemSize);
EvaluateAgainstObjectiveFunctions(Population);
FastNondominatedSort(Population);
Selected  $\leftarrow$  SelectParentsByRank(Population,  $Population_{size}$ );
Children  $\leftarrow$  CrossoverAndMutation(Selected,  $P_{crossover}$ ,  $P_{mutation}$ );
while  $\neg$ StopCondition() do
    EvaluateAgainstObjectiveFunctions(Children);
    Union  $\leftarrow$  Merge(Population, Children);
    Fronts  $\leftarrow$  FastNondominatedSort(Union);
    Parents  $\leftarrow \emptyset$ ;
     $Front_L \leftarrow \emptyset$ ;
    foreach  $Front_i \in$  Fronts do
        CrowdingDistanceAssignment( $Front_i$ );
        if Size(Parents)+Size( $Front_i$ ) >  $Population_{size}$  then
             $Front_L \leftarrow i$ ;
            Break();
        else
            Parents  $\leftarrow$  Merge(Parents,  $Front_i$ );
        end
    end
    if Size(Parents) <  $Population_{size}$  then
         $Front_L \leftarrow$  SortByRankAndDistance( $Front_L$ );
        for  $P_1$  to  $P_{Population_{size}-Size(Front_L)}$  do
            Parents  $\leftarrow P_i$ ;
        end
    end
    Selected  $\leftarrow$  SelectParentsByRankAndDistance(Parents,  $Population_{size}$ );
    Population  $\leftarrow$  Children;
    Children  $\leftarrow$  CrossoverAndMutation(Selected,  $P_{crossover}$ ,  $P_{mutation}$ );
end
return Children;

```

Додаток Б
ПОРІВНЯННЯ АЛГОРИТМІВ NSGA-II ТА NSGA-II-UPCA

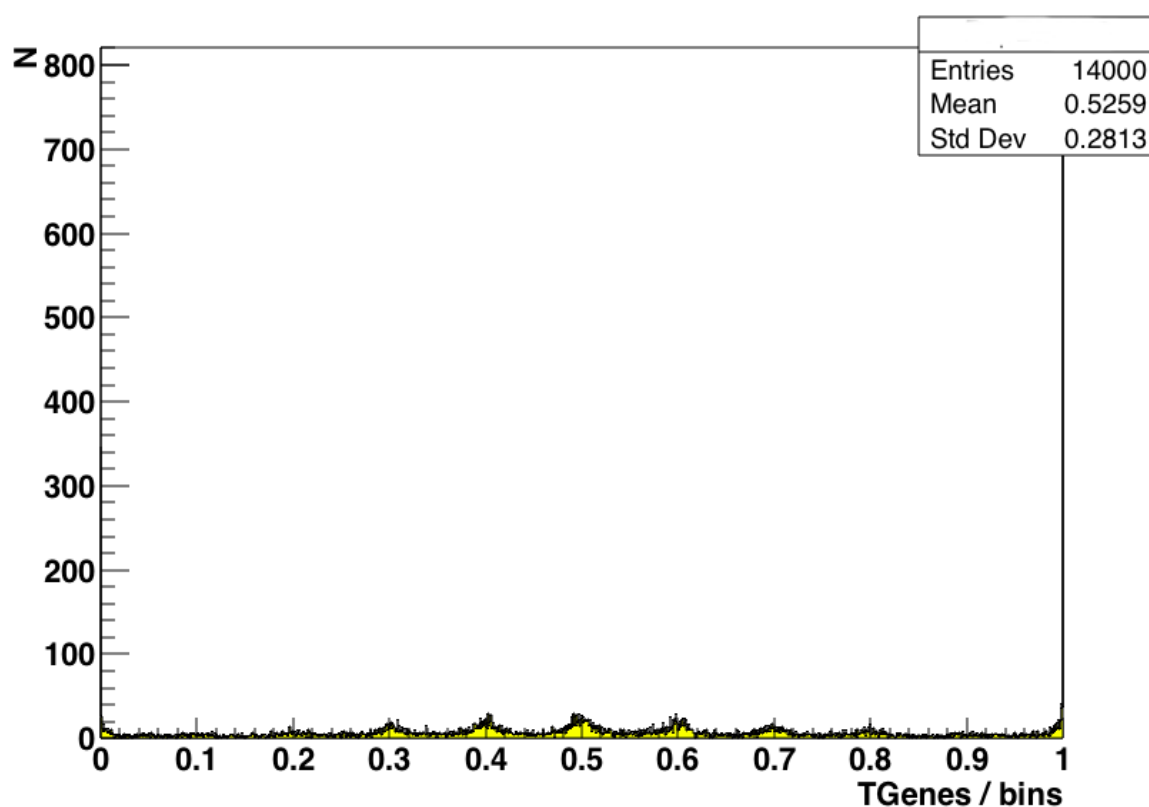


Рисунок Б.1 – Гістограма розподілу генетичної популяції для 100-го покоління для алгоритму NSGA-II та тесту DTLZ1

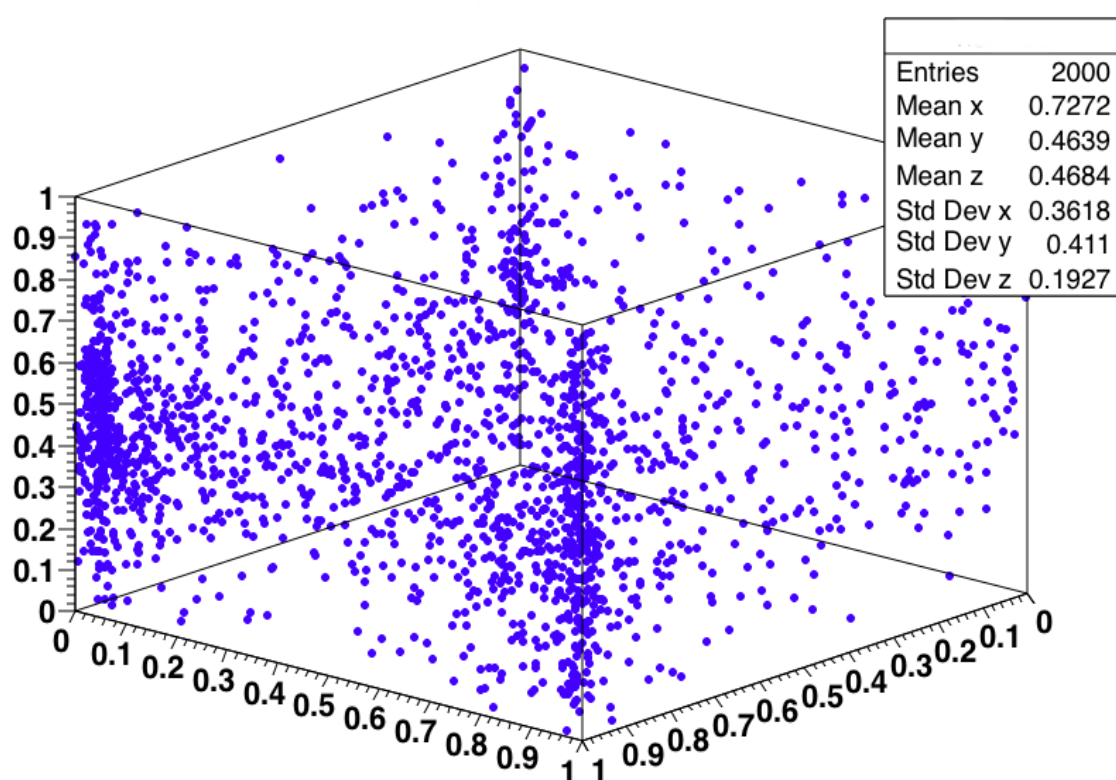


Рисунок Б.2 – Розподіл параметрів для 100-го покоління для алгоритму NSGA-II та тесту DTLZ1

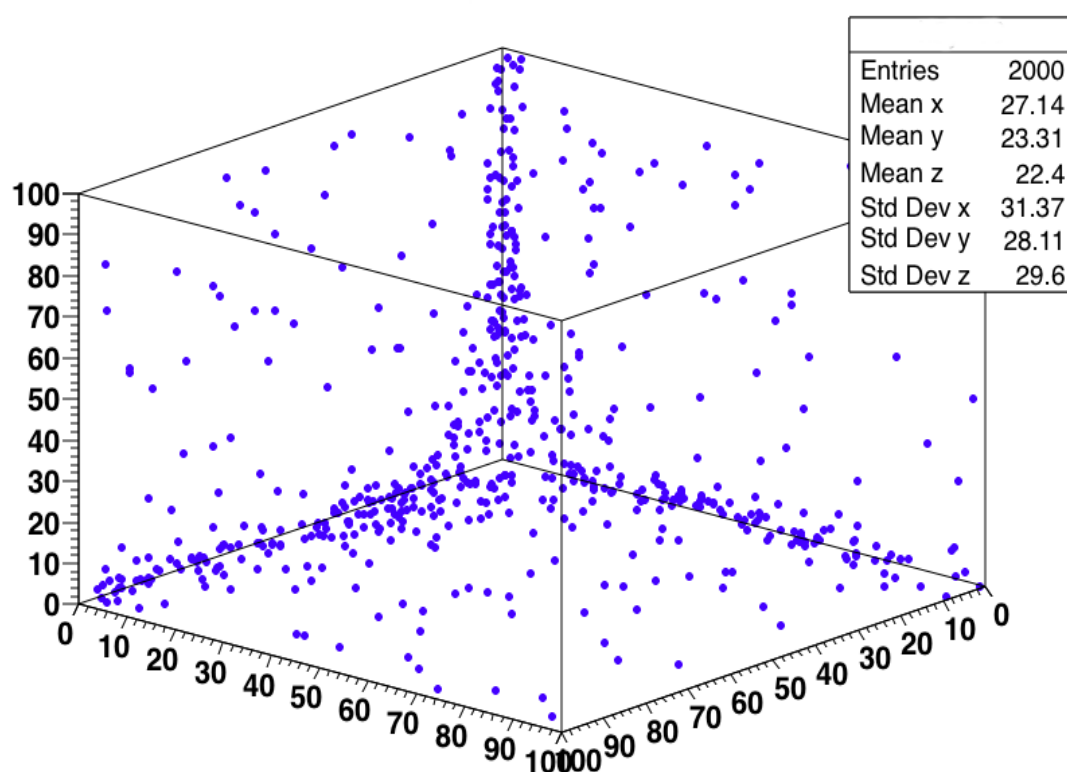


Рисунок Б.3 – Розподіл значень функції вартості генетичного алгоритму для 100-го покоління для алгоритму NSGA-II та тесту DTLZ1

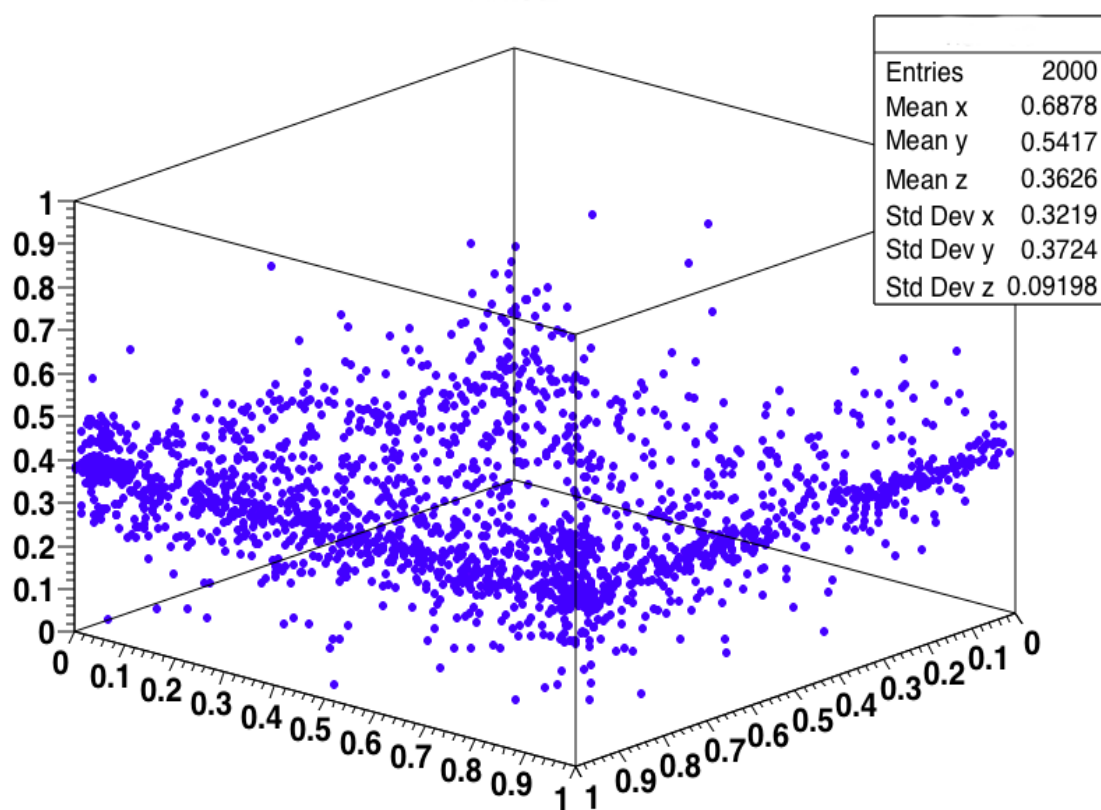


Рисунок Б.4 – Розподіл параметрів для 100-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ1

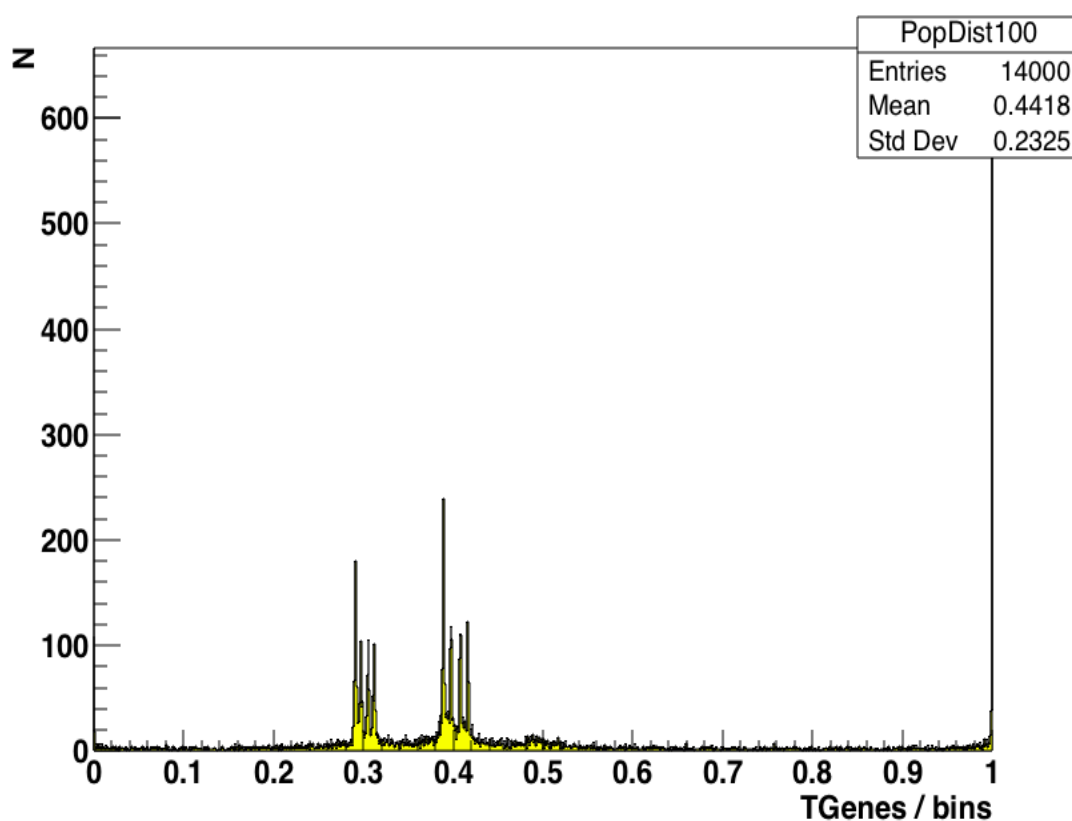


Рисунок Б.5 – Гістограма розподілу генетичної популяції для 100-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ1

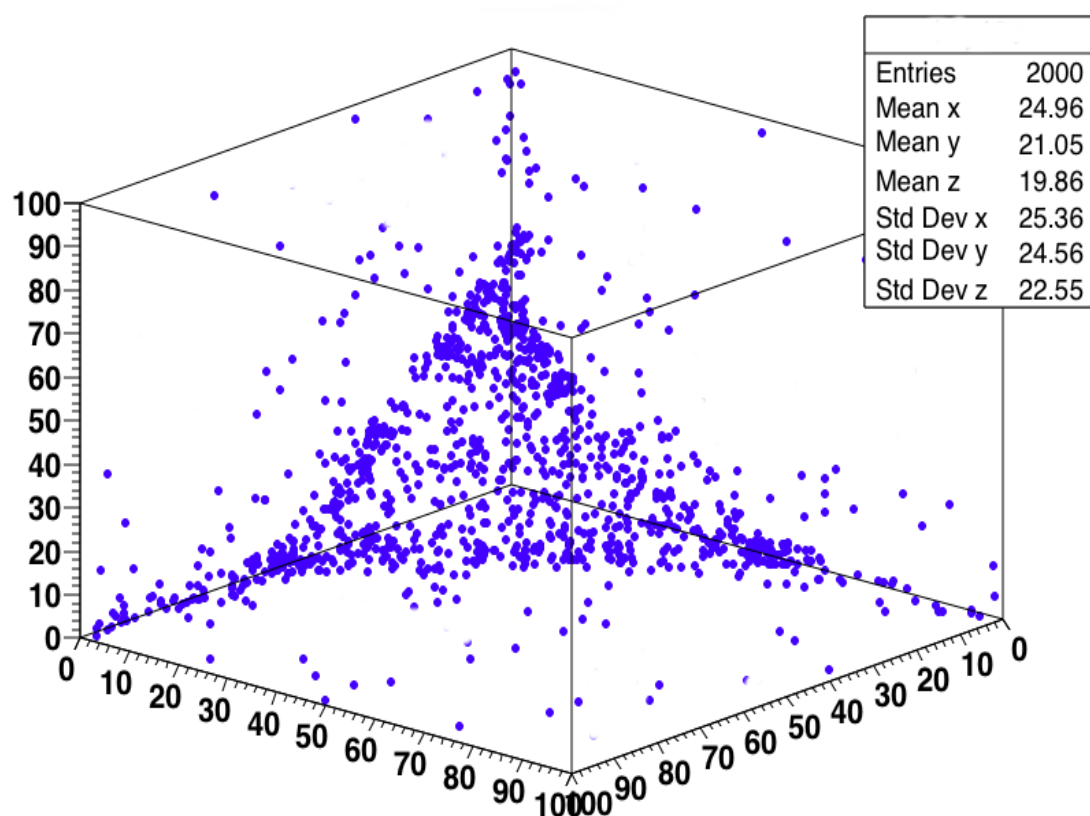


Рисунок Б.6 – Розподіл значень функції вартості генетичного алгоритму для 100-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ1

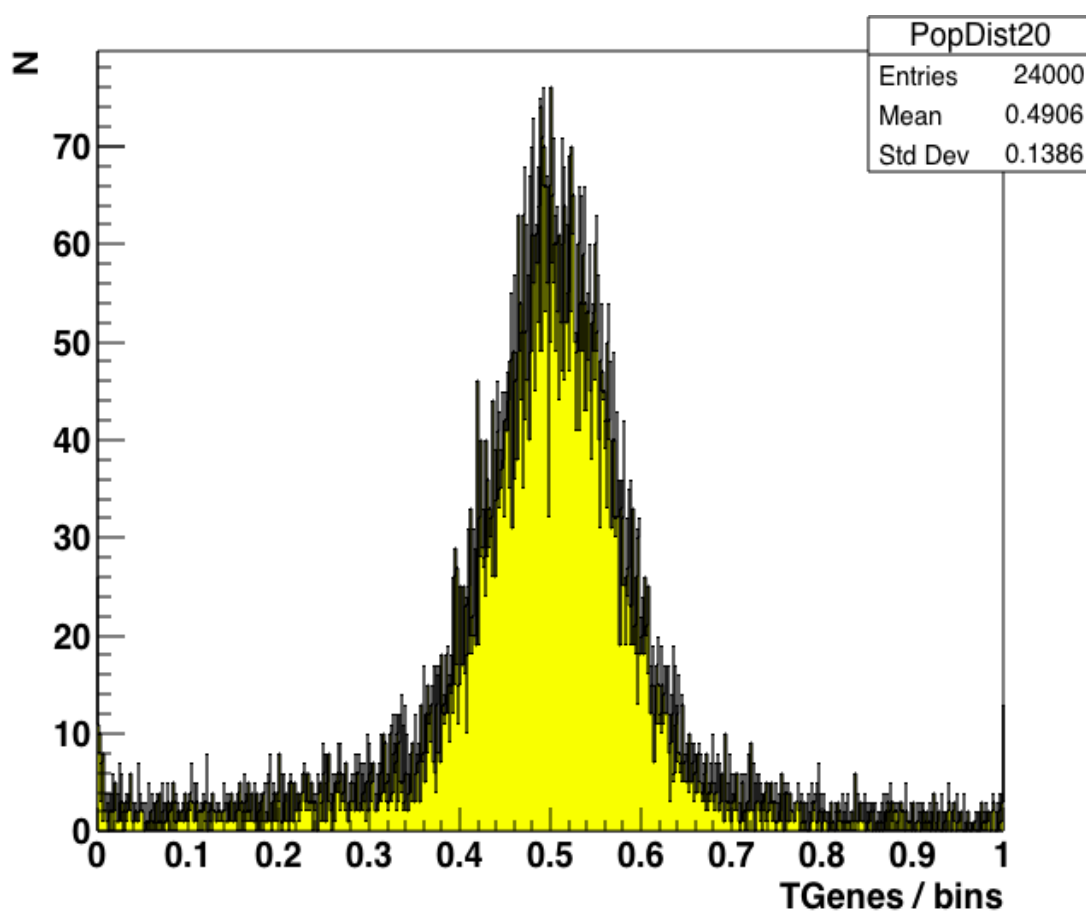


Рисунок Б.7 – Гістограма розподілу генетичної популяції для 200-го покоління для алгоритму NSGA-II та тесту DTLZ2

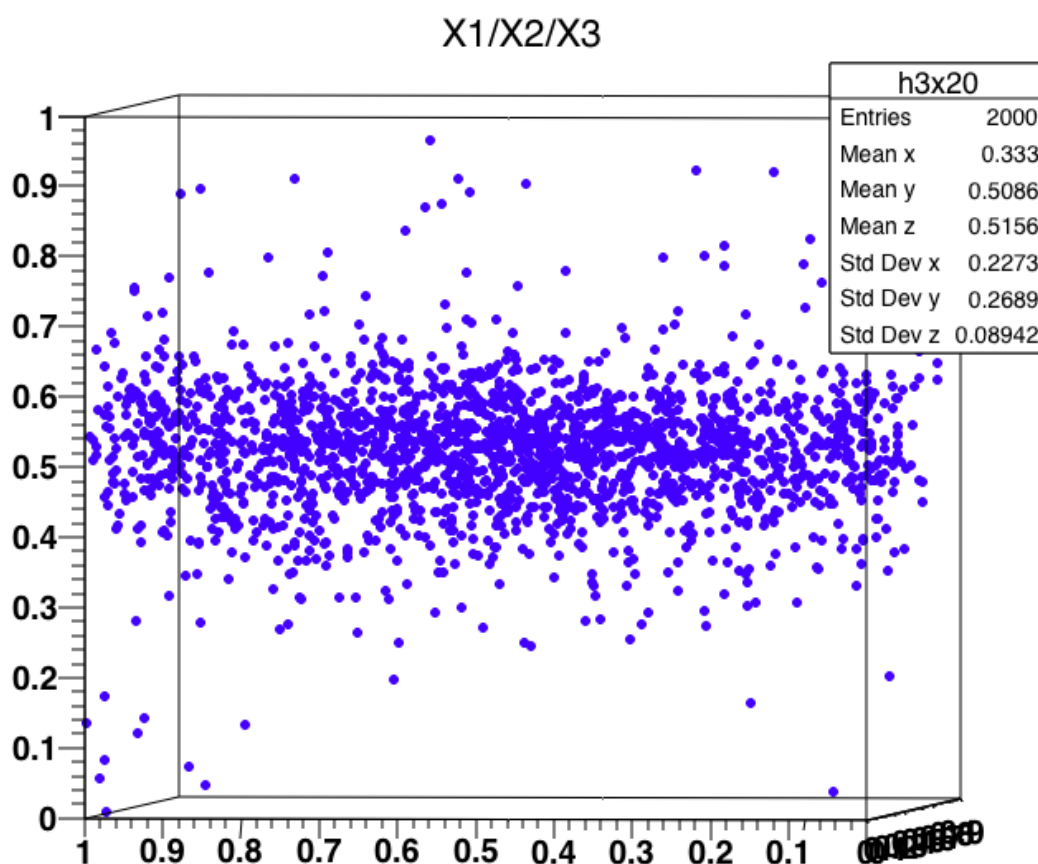


Рисунок Б.8 – Розподіл параметрів для 200-го покоління для алгоритму NSGA-II та тесту DTLZ2

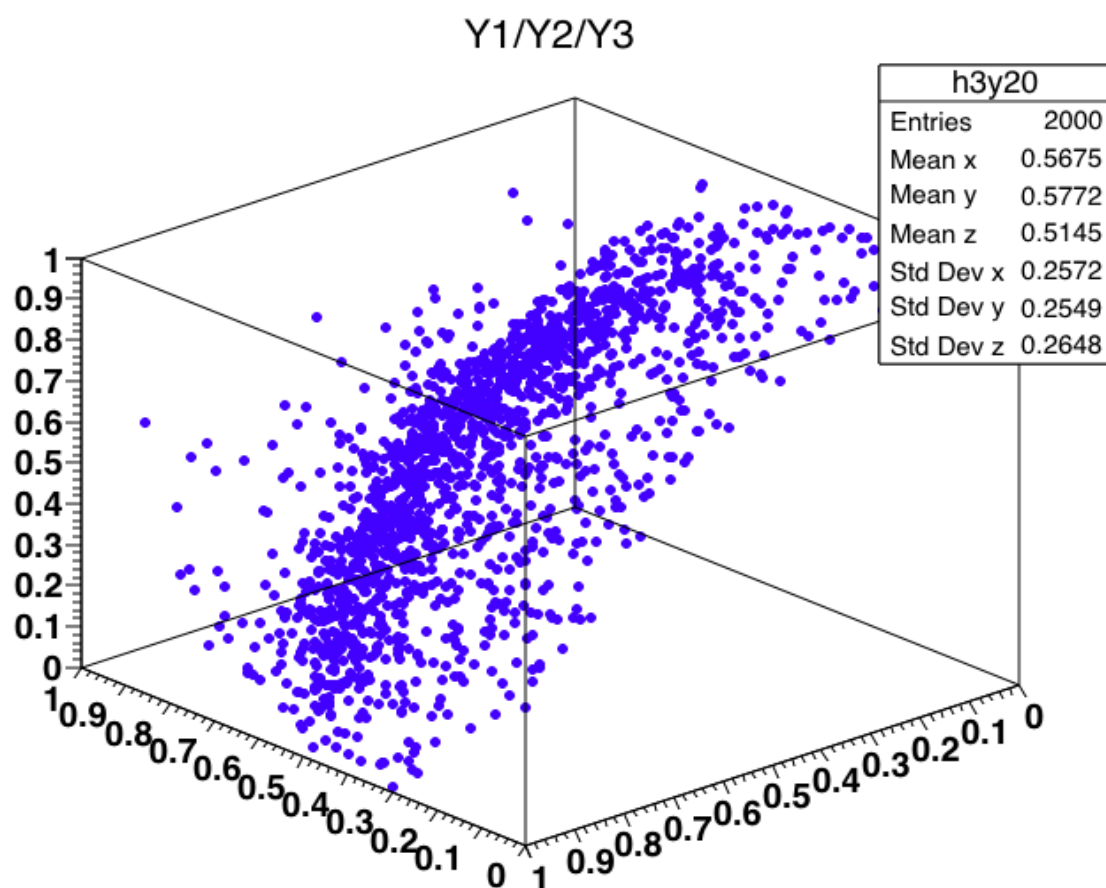


Рисунок Б.9 – Розподіл значень функції вартості генетичного алгоритму для 200-го покоління для алгоритму NSGA-II та тесту DTLZ2

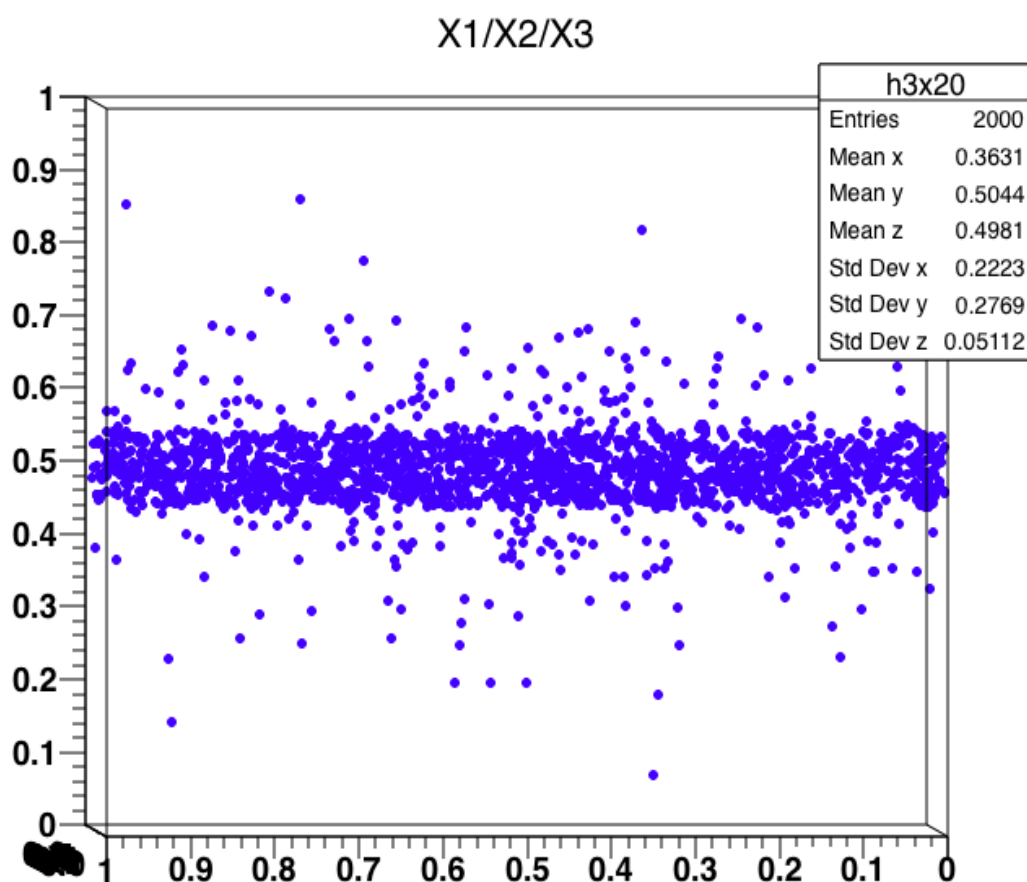


Рисунок Б.10 – Залежність X1/X2/X3 для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ3

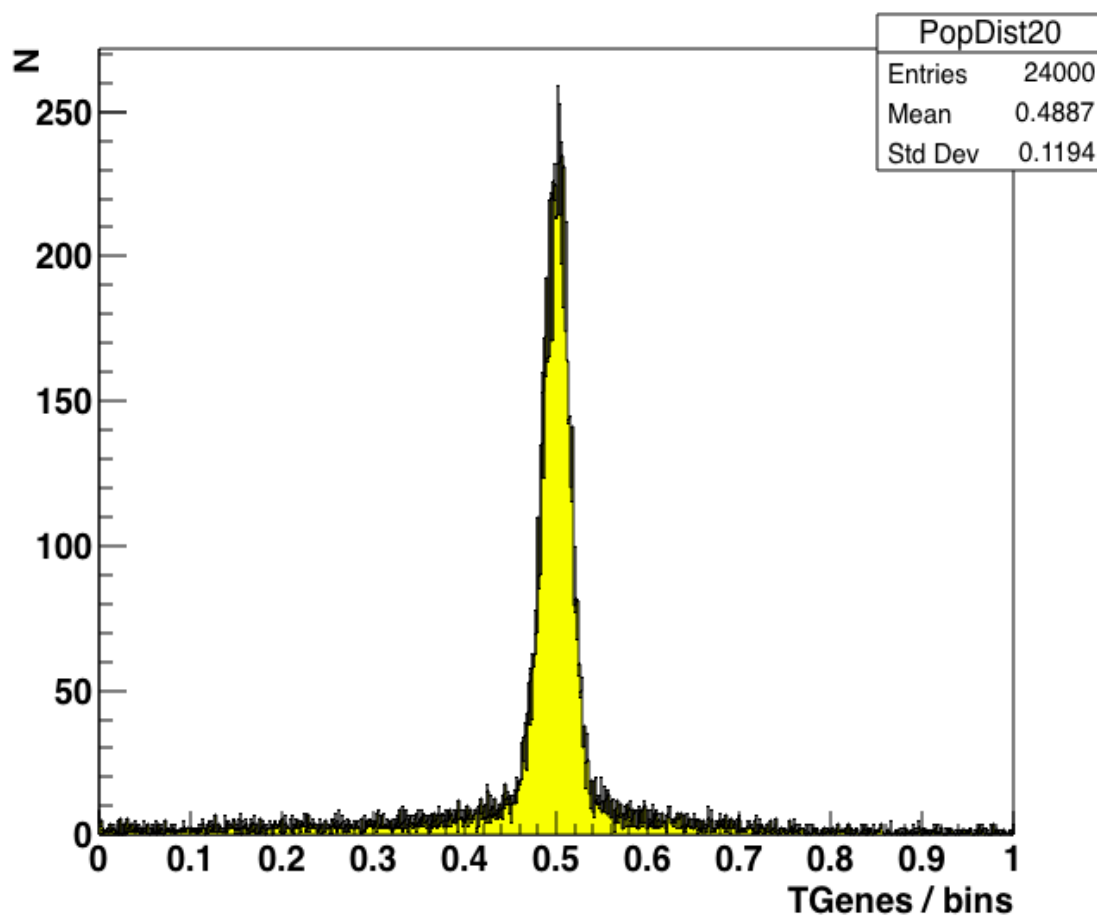


Рисунок Б.11 – Гістограма розподілу генетичної популяції для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ2

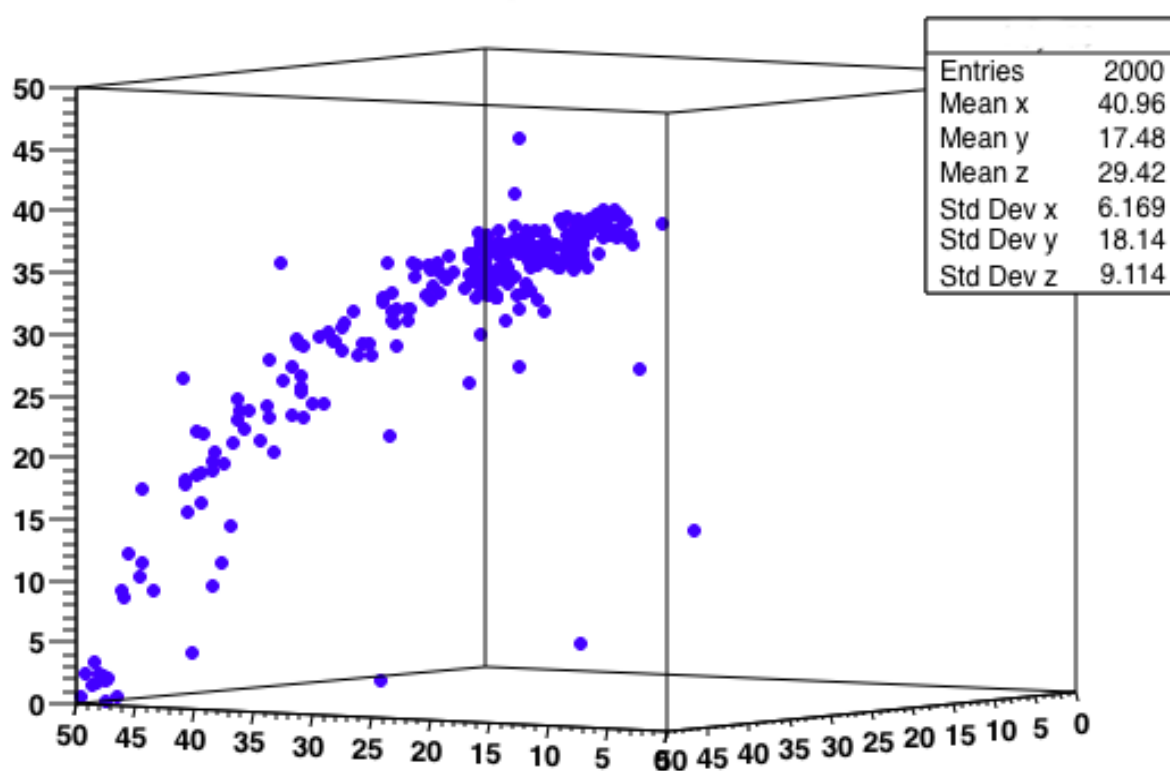


Рисунок Б.12 – Розподіл значень функції вартості генетичного алгоритму для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ3

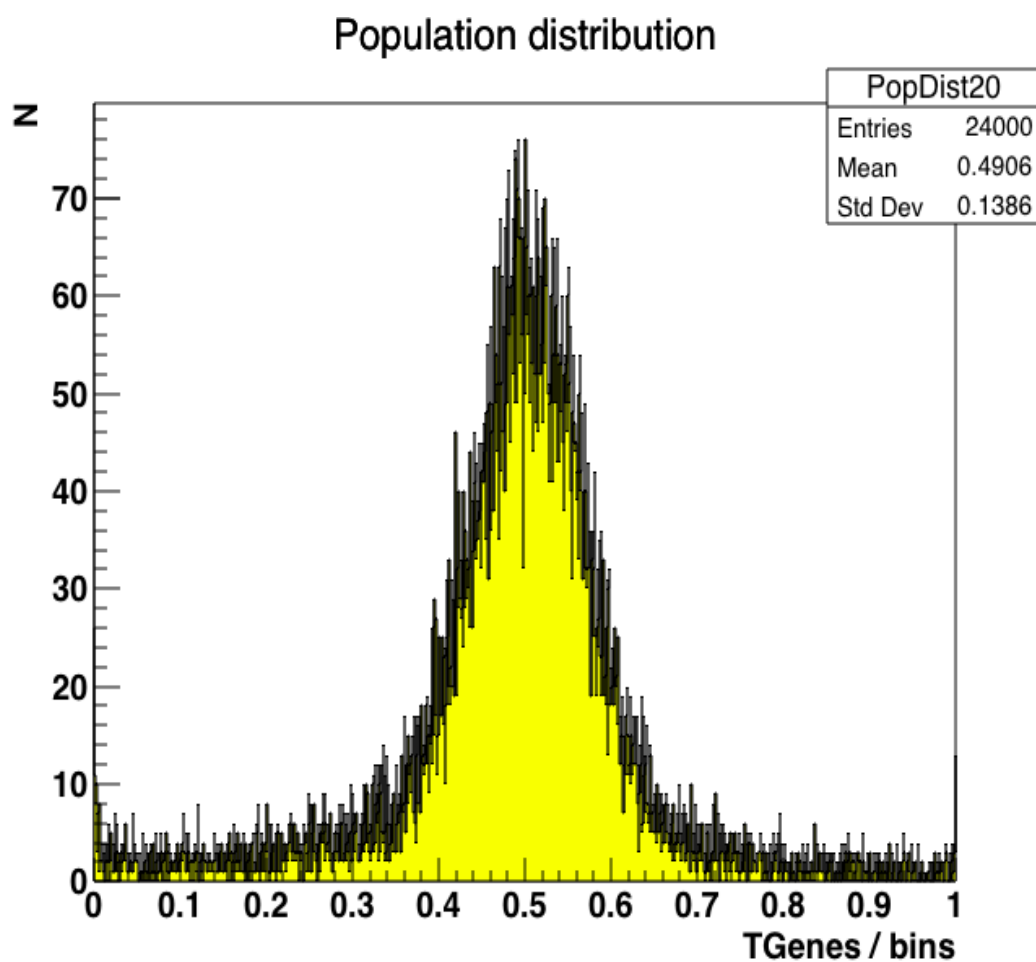


Рисунок Б.13 – Гістограма розподілу генетичної популяції для 200-го покоління для алгоритму NSGA-II та тесту DTLZ2

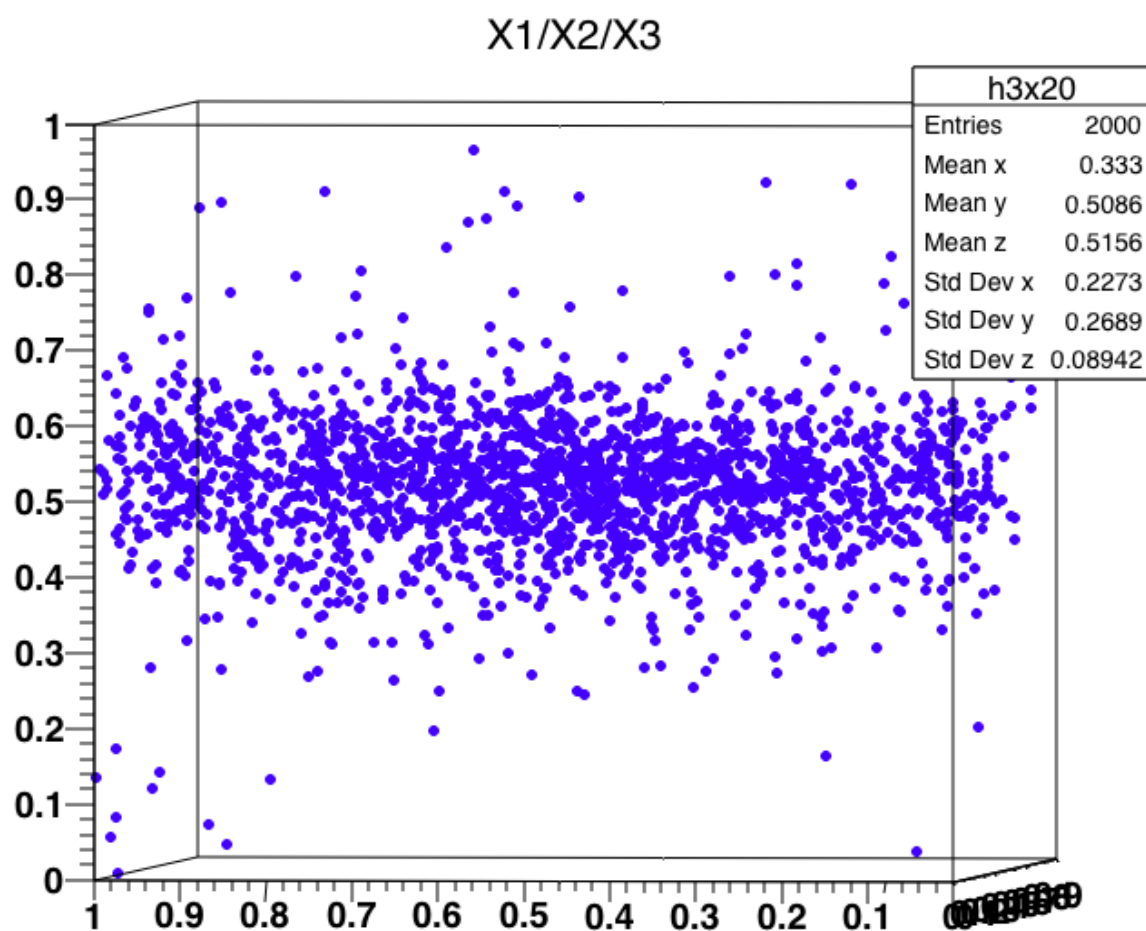


Рисунок Б.14 – Залежність параметрів для 200-го покоління для алгоритму NSGA-II та тесту DTLZ2

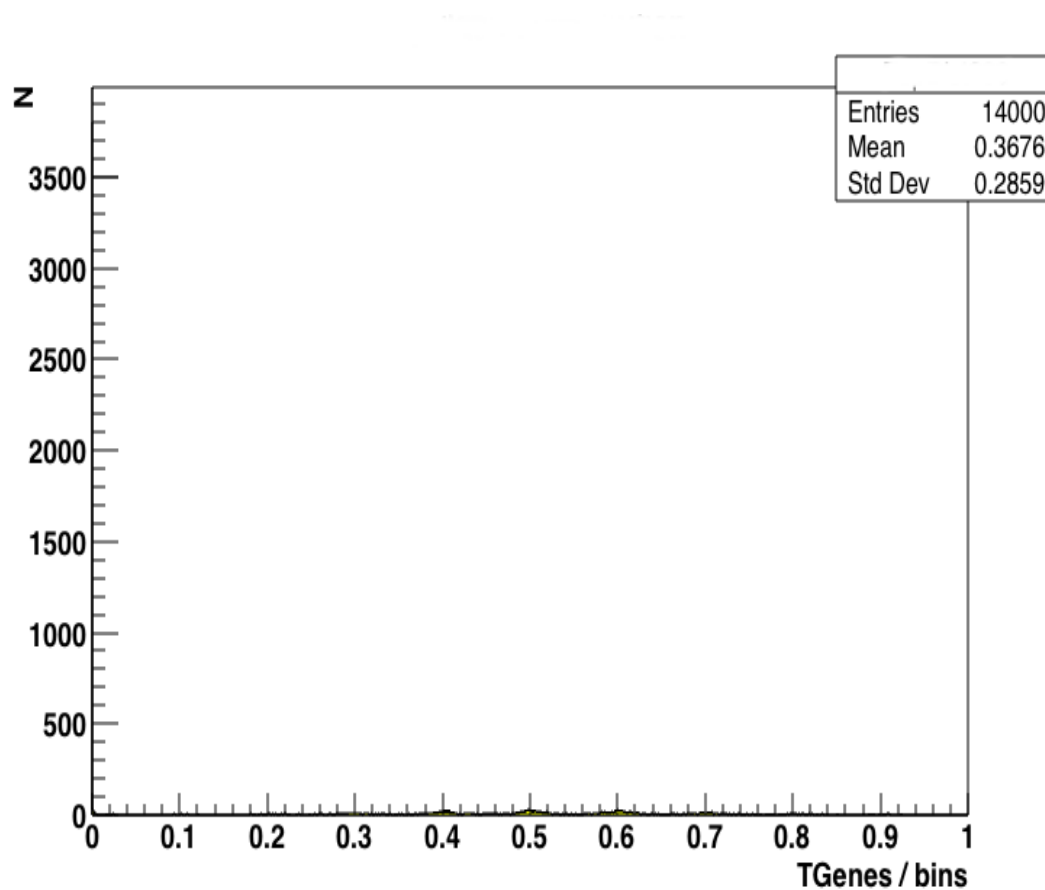


Рисунок Б.15 – Розподіл значень функції вартості генетичного алгоритму для 200-го покоління для алгоритму NSGA-II та тесту DTLZ3

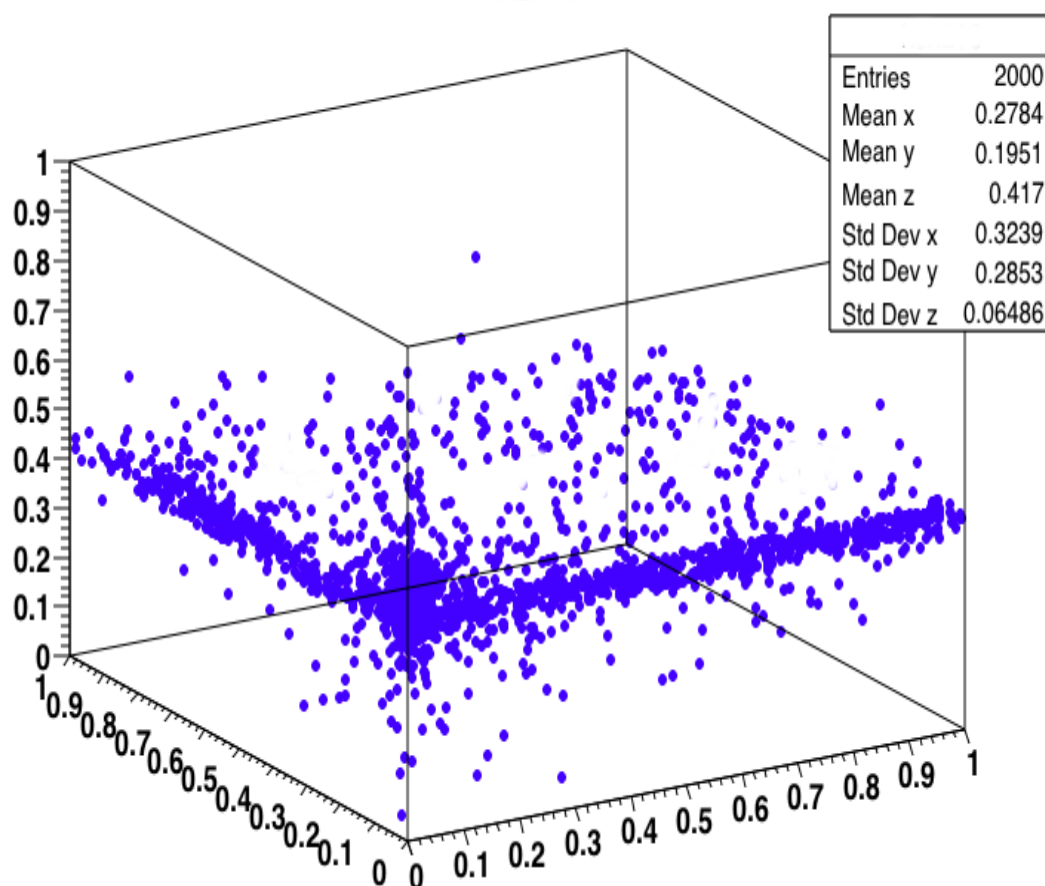


Рисунок Б.16 – Залежність $X1/X2/X3$ для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ3

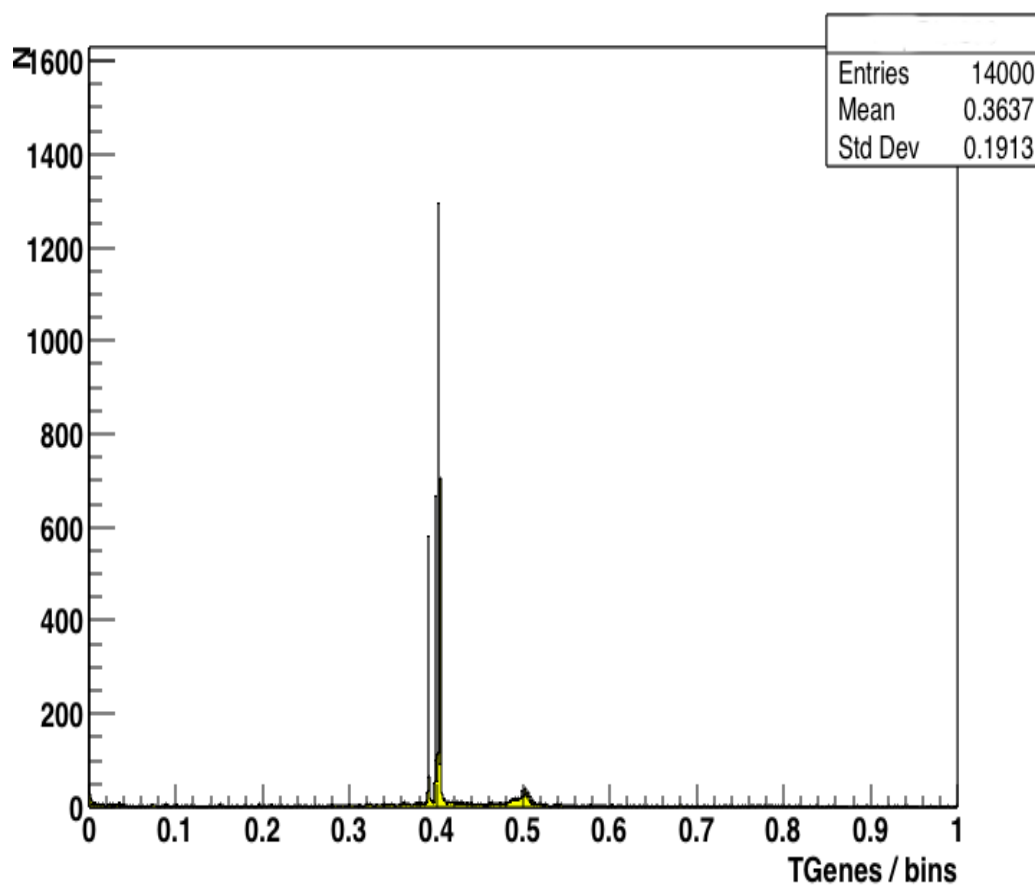


Рисунок Б.17 – Гістограма розподілу генетичної популяції для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ3

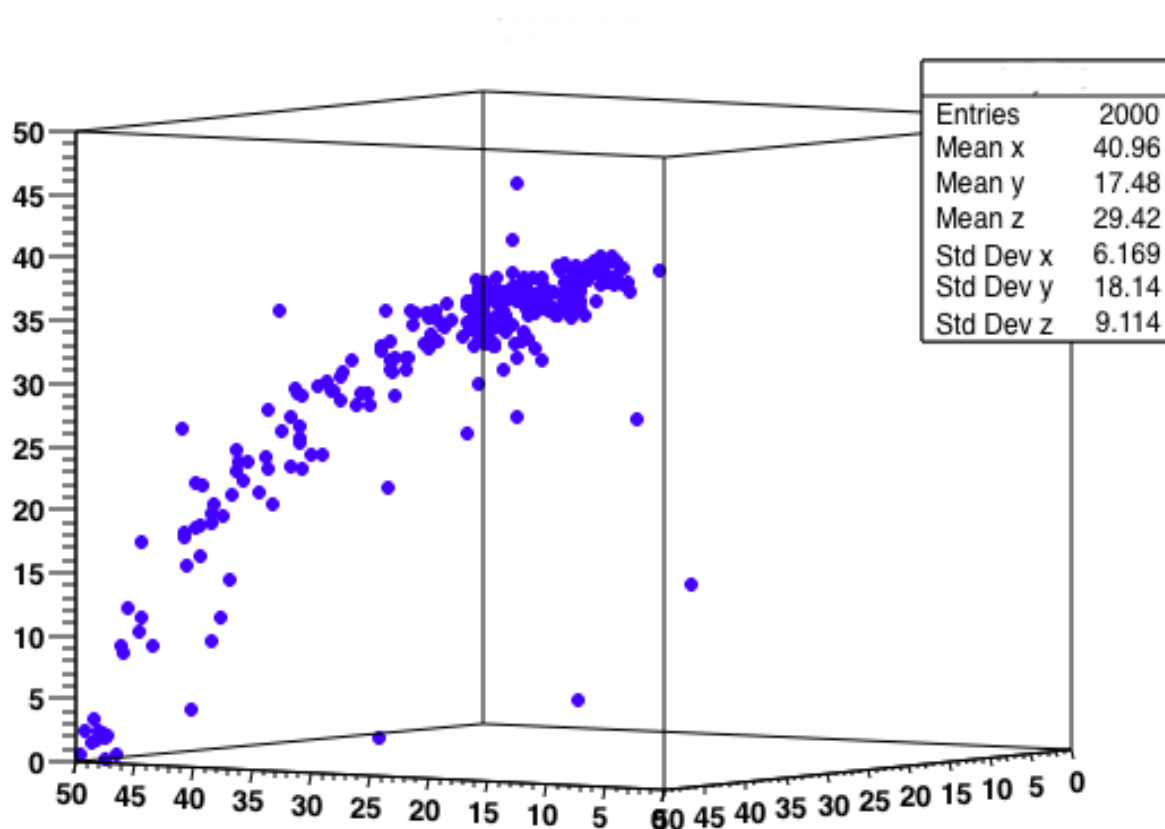


Рисунок Б.18 – Розподіл значень функції вартості генетичного алгоритму для 200-го покоління для алгоритму NSGA-II з попередньою обробкою даних (UPCA) та тесту DTLZ3

Додаток В

АКТИ ВПРОВАДЖЕННЯ

«ЗАТВЕРДЖУЮ»

Заступник директора Інституту
прикладного системного аналізу
КПІ ім. Ігоря Сікорського
проф. д.т.н.



Романенко В.Д.

«20» жовтня 2019 р.

ДОВІДКА

про використання результатів дисертаційної роботи Шадури Оксани
Віталіївни
у навчальному процесі Інституту прикладного системного аналізу
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»

Результати дисертаційного дослідження Шадури Оксани Віталіївни за темою «Стохастична оптимізація продуктивності пакету симуляції транспорту частинок в фізиці високих енергій», представленого на здобуття наукового ступеня кандидата технічних наук, спрямовані на аналіз та оптимізацію продуктивності програмних додатків в фізиці високих енергій і пропонують перспективні методи та технології для високопродуктивних обчислень в грид-хмарному середовищі.

Отримані результати впроваджено в навчальний процес кафедри системного проектування у вигляді методик програмування і моделювання.

В курсі лекцій «Розподілені високопродуктивні обчислювальні системи» розглядається тема: «Структурні елементи та алгоритми багатокритеріальної задачі оптимізації».

Учений секретар кафедри
системного проектування

О.Ю. Безносик

“ЗАТВЕРДЖУЮ”

Директор

Інституту теоретичної фізики

ім.М.М.Боголюбова НАН України,

академік НАН України,

Загородний А.Г.



« 9 » жовтня 2019р.

АКТ

впровадження результатів дисертаційного дослідження провідного інженера Інституту теоретичної фізики ім.М.М.Боголюбова НАН України **Шадури Оксани Віталіївни** на тему «**Стохастична оптимізація продуктивності пакету симуляції транспорту частинок в фізиці високих енергій**» на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.12 – Системи автоматизації проектувальних робіт.

Комісія у складі:

Голова комісії – заступник директора Інституту теоретичної фізики ім.М.М.Боголюбова НАН України кандидат фізико - математичних наук **Глушко Микола Іванович**,

Члени комісії: головний науковий співробітник відділу високих густин енергій Інституту теоретичної фізики ім.М.М.Боголюбова НАН України, доктор фізико - математичних наук **Синюков Юрій Михайлович**, завідувач відділу комп'ютерного забезпечення наукових досліджень і науково-технічної інформації Інституту теоретичної фізики ім.М.М.Боголюбова НАН України, кандидат технічних наук, – **Свістунів Сергій Якович**,

цим Актом засвідчує, що результати дисертаційного дослідження Шадури О.М. на тему «Стохастична оптимізація продуктивності пакету симуляції транспорту частинок в фізиці високих енергій», а саме:

а) модифікація генетичного алгоритму що розроблена за допомогою запропонованого в дисертаційному дослідженні нового генетичного оператора на основі формулювання математичної моделі методу нецентрованих головних компонент та знаходження оцінки наближень при застосуванні цього методу для аналізу даних;

б) розроблений метод включення нового генетичного оператора в типовий генетичний алгоритм та дослідження ефективності його застосування на прикладі еталонних тестів для багатокритеріальних задач, якій забезпечив прискорення збіжності генетичного алгоритму до оптимального Парето-фронту і тим самим дозволив отримати наближення рішення для задач багатокритеріальної оптимізації,

були перевірені при застосованні модифікації генетичного алгоритму для еталонних тестів багатокритеріальних задач і показали, що загальний час виконання партій симуляцій за допомогою модифікованого генетичного алгоритму скоротився приблизно на 18 % (для процесорів Intel(R) Core i7) і на 27% з одночасним зменшенням часу використання CPU на 34% (для процесорів Intel(R) Xeon(R) E5-2695).

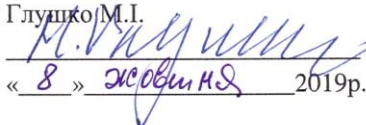
Модифікація генетичного алгоритму була реалізована в якості програмного модулю і впроваджена в бібліотеці програмних модулів GeantV, яка використовується для оптимізації продуктивності програмового пакету симуляцій транспорту частинок GeantV для обробки даних в експериментах на Великому адронному колайдері в ЦЕРН. Використання нової модифікації генетичного алгоритму для оптимізації роботи пакету GeantV для моделювання транспорту частинок при обробці експериментальних даних дозволило скоротити час виконання симуляцій в середньому від 9% до 15% в залежності від набору даних в порівнянні з використанням програмного пакету GeantV без модулю оптимізації.

Дисертаційне дослідження Шадури О.М. на тему «Стохастична оптимізація продуктивності пакету симуляції транспорту частинок в фізиці високих енергій» виконано в рамках Угоди про співробітництво Інституту теоретичної фізики ім.М.М.Боголюбова НАН України та колаборації ALICE в ЦЕРН, Женева, Швейцарія, та проєктів Інституту теоретичної фізики ім.М.М.Боголюбова НАН України в рамках виконання робіт за Цільовою комплексною програмою наукових досліджень НАН України «Грид-інфраструктура і грид-технології для наукових і науково-прикладних застосувань» в 2016–2018 рр. (державний реєстраційний номер № 0116U004355, 0117U003430).

Голова комісії:

кандидат фізико - математичних наук

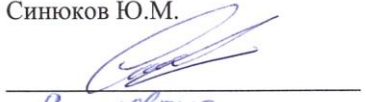
Глушко М.І.


« 8 » жовтня 2019р.

Члени комісії:

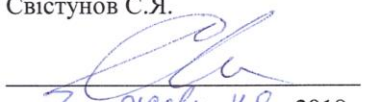
доктор фізико - математичних наук

Синюков Ю.М.


« 9 » жовтня 2019р.

кандидат технічних наук

Свістунов С.Я.


« 7 » жовтня 2019р.

Додаток Г

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА
ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ*Основні публікації*

1. Shadura O. et al. Multivariate convergence-targeted operator for the genetic algorithm / O. Shadura, A. Petrenko, S. Svistunov // Системні дослідження та інформаційні технології. – 2017. – № 1. – С. 126–140. – Бібліогр.: 17 назв.

Журнал включено в такі міжнародні наукометричні бази: Index Copernicus, DOAJ, EBSCOhost Reaserch Database, особистий внесок здобувача – розроблено математичний апарат методу нецентрованих головних компонент (НГК).

2. Оксана Шадура. «Метод головних компонент і оптимізація пакетів фізичного моделювання за допомогою генетичних алгоритмів» // Вісник Університету «Україна», Серія «Інформатика, обчислювальна техніка та кібернетика». – 2019. – №1 (22) – С. 198–209. – Бібліогр.: 16 назв.

Особистий внесок здобувача – проведений аналіз вибору оптимальної модифікації генетичного алгоритму за рахунок включення в алгоритм НГК-оператора та досліджена продуктивність такого генетичного алгоритму.

3. Оксана Шадура. Модифікація генетичних алгоритмів на основі методу нецетрованих головних компонент та стандартні тести, // World Science – 2019. – № 4(44). – С. 4–11. – Бібліогр.: 10 назв. – DOI: 10.31435/rsglobal_ws.

Журнал включено в такі міжнародні наукометричні бази: Index Copernicus, Google Scholar, academia.edu. Особистий внесок здобувача

– зформульовано підхід до оптимізації програмного пакету GeantV.

4. Oksana Shadura et al. Performance Optimization of Physics Simulations Through Genetic Algorithms. / Oksana Shadura, Federico Carminati and Anatoliy Petrenko. // Journal of Computer Science. – 2019. – Volume 15. – Issue 1. – P. 5766. – DOI 10.3844/jcssp.2019.57.66.

Журнал включено в такі міжнародні наукометричні бази: Scopus, EBSCO, особистий внесок здобувача – було досліджено вибір ключових параметрів програмного пакету GeantV для оптимізації продуктивності цього пакету та перевірено цей набір параметрів.

5. Оксана Шадура. Оптимізація пакету фізичного моделювання GeantV за допомогою генетичних алгоритмів. // Вчені записки Таврійського національного університету імені В. І. Вернадського, Серія: Технічні науки – 2019. – Том 30 (69). – № 3. – С. 206–213.

Журнал включено в такі міжнародні наукометричні бази: Index Copernicus, особистий внесок здобувача – було показано, що модифікований генетичний алгоритм з включенням НГК-оператора прискорює оптимізацію продуктивності програмного пакету GeantV при симуляції транспорту частинок через простий калориметр та збільшує ефективність використання обчислювальних ресурсів для аналізу даних у фізиці високих енергій.

Апробація результатів дисертації

6. O.Shadura et al. The experimental facility for the Search for Hidden Particles at the CERN SPS / Ahdida C., Albanese R., ..., Shadura O., ... , Zaytsev Yu // Journal of Instrumentation (JINST). – 2019. – Volume 14. – Issue 3. – P. 3025–3046 – DOI 10.1088/1748-0221/14/03/p03025

Особистий внесок здобувача – було проведено аналіз слабких та

сильних сторін програмного пакету Geant4, який є попередником пакету нового покоління.

7. Oksana Shadura. Stochastic performance tuning of complex simulation applications using unsupervised machine learning. / Oksana Shadura, Federico Carminati // 2016 IEEE Symposium Series on Computational Intelligence – 2016. – ISBN 978-1-5090-4240-1.

Особистий внесок здобувача – розроблено підхід до оптимізації програмного пакету GeantV.

8. G. Amadio and all. Stochastic optimization of GeantV code by use of genetic algorithms / G. Amadio and J. Apostolakis and M. Bandieramonte and S.P. Behera and R. Brun and P. Canal and F. Carminati and G. Cosmo and L. Duhem and D. Elvira and G. Folger and A. Gheata and M. Gheata and I. Goulas and F. Hariri and S.Y. Jun and D. Konstantinov and H. Kumawat and V. Ivantchenko and G. Lima and T. Nikitina and M. Novak and W. Pokorski and A. Ribon and R. Seghal and O. Shadura and S. Vallecorsa and S. Wenzel, Stochastic optimization of GeantV code by use of genetic algorithms // Journal of Physics: Conf. Ser. Offline Computing – 2017. – Vol. 898, N. 4, – P. 042026 – IOP Publishing.

Особистий внесок здобувача – розроблена імплементація системи оптимізації програмного пакету GeantV.

9. Amadio G and all. GeantV: from CPU to accelerators / G. Amadio, A. Ananya, J. Apostolakis, A. Arora, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F. Carminati, L. Duhem, D. Elvira, A. Gheata, M. Gheata, I. Goulas, R. Iope, S. Jun, G. Lima, A. Mohanty, T. Nikitina, M. Novak, W. Pokorski, A. Ribon, R. Sehgal, O. Shadura, S. Vallecorsa, S. Wenzel, Y. Zhang // Journal of Physics: Conf. Ser. Computing Technology for Physics Research – 2016. – Vol. 762, N. 1. – P. 012019. – IOP Publishing.

Особистий внесок здобувача – опис термінології пакету GeantV, що визначає основну проблематику та можливі алгоритми, що мають бути дослідженні для вирішення проблеми оптимізації.

10. Amadio G and all. The GeantV project: preparing the future of simulation / G. Amadio, J. Apostolakis, M. Bandieramonte, A. Bhattacharyya, C. Bianchini, R. Brun, P. Canal, F Carminati, J. C. De Fine Licht, L. Duhem, V. D. Elvira, A. Gheata, R. L. Iope, G. Lima, A. Mohanty, T. Nikitina, M. Novak, W. Pokorsky, R. Sehgal, O. Shadura, S. Vallecorsa, S. Wenzel // Journal of Physics: Conf. Ser. Offline software – 2015. – Vol. 664, N. 7. – P. 072006 – IOP Publishing.

Особистий внесок здобувача – робота над системою паралелізму у пакеті GeantV та інтеграцією системи оптимізації в пакет GeantV.

11. Apostolakis J and all. Adaptive track scheduling to optimize concurrency and vectorization in GeantV / J. Apostolakis, M. Bandieramonte, G. Bitzes, R. Brun, P. Canal, F Carminati, J. C. De Fine Licht, L. Duhem, V. D. Elvira, A. Gheata, S. Y. Jun, G. Lima, M. Novak, R. Sehgal, O. Shadura, S. Wenzel // Journal of Physics: Conf. Ser. Advanced Computing and Analysis Techniques in Physics Research – 2015. – Vol. 608, N 1. – P. 012003 – IOP Publishing.

Особистий внесок здобувача – визначення основної проблематики для оптимізації пакету GeantV та параметрів, що мають бути дослідженні для оптимізації пакету GeantV.

12. J. Apostolakis and all. Towards a high performance geometry library for particle–detector simulations // J. Apostolakis and M. Bandieramonte and G. Bitzes and R. Brun and P. Canal and F. Carminati and G. Cosmo and J. C. De Fine Licht and L. Duhem and V. D. Elvira and A. Gheata and S. Y. Jun and G. Lima and T. Nikitina and M. Novak and R. Sehgal and O. Shadura and S. Wenzel // Journal of Physics: Conf. Ser. Computing Technology for Physics Research – 2015. – Vol.

608, N. 1, – P. 012023 – IOP Publishing.

Особистий внесок здобувача – визначення основної проблематики для оптимізації пакету VecGeom та можливих алгоритмів та параметрів, що мають бути досліджені для вирішення проблеми оптимізації параметрів пакету GeantV.